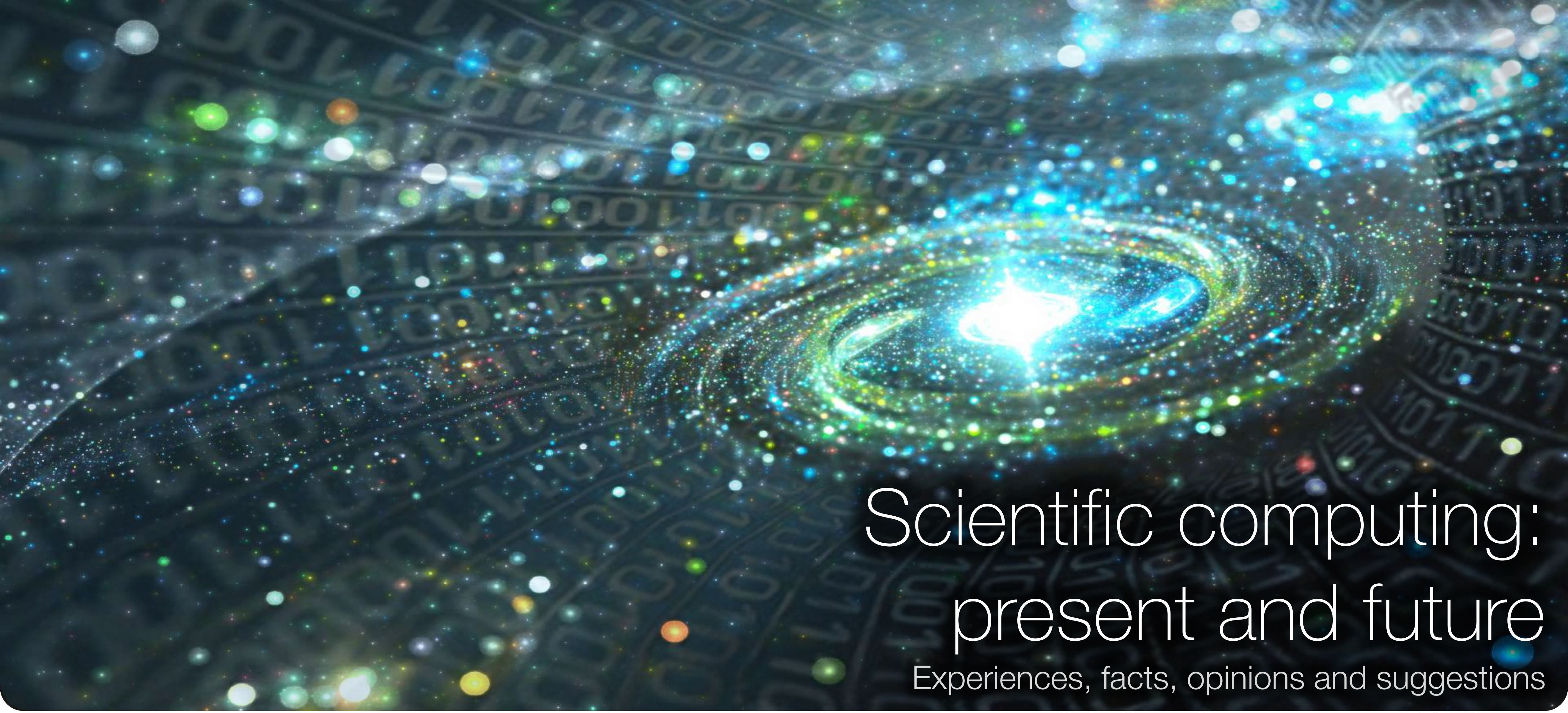


Scientific computing: present and future

Experiences, facts, opinions and suggestions

Seminario Gruppo 2
9th March 2021

Istituto Nazionale di Fisica Nucleare

Gabriele Gaetano Fronzé
for the INFN-Torino Computing Group

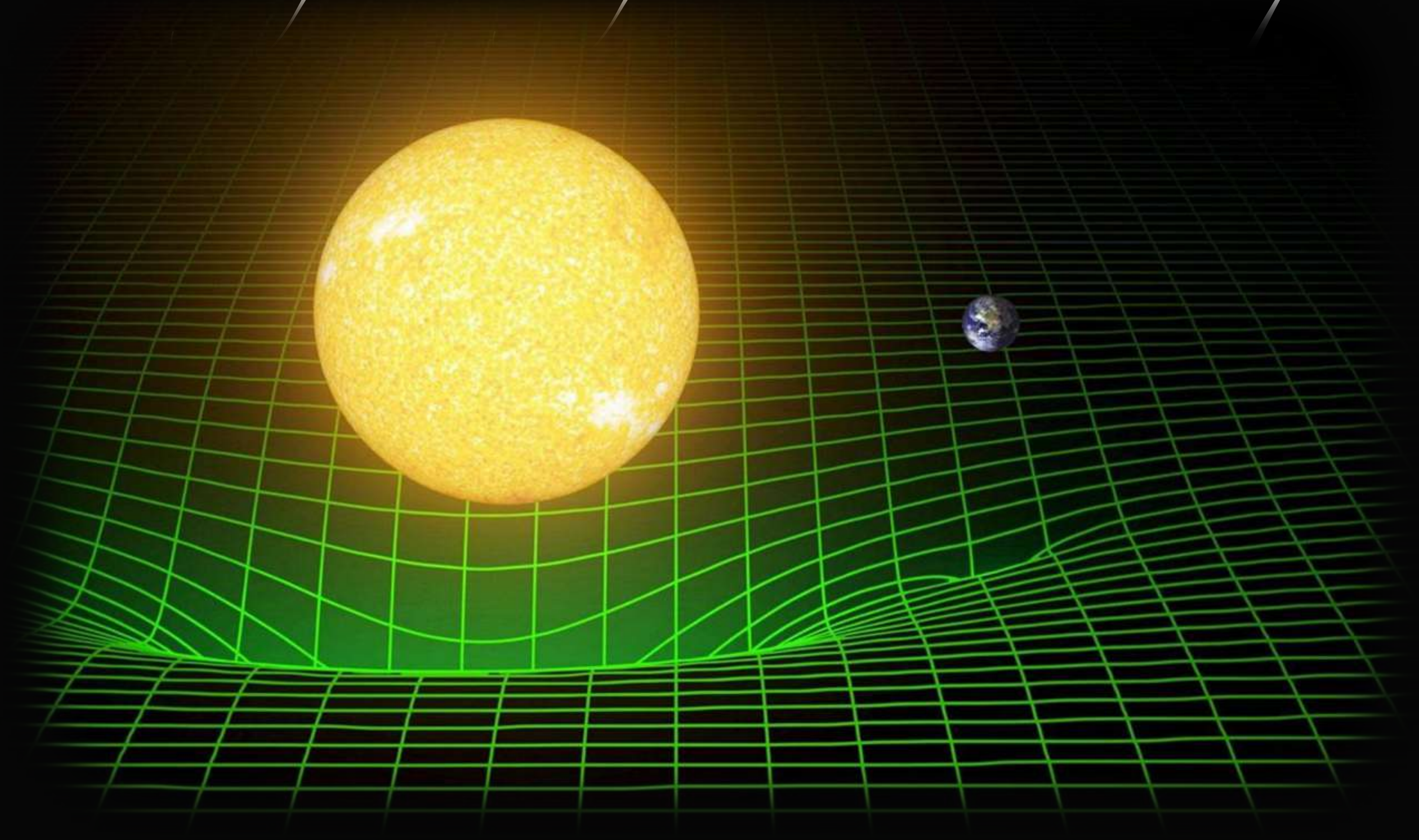
Summary

- Gravitational waves for dummies
- (Scientific) computing assets
- Virgo computing architecture
- The Outer World
- Conclusions?

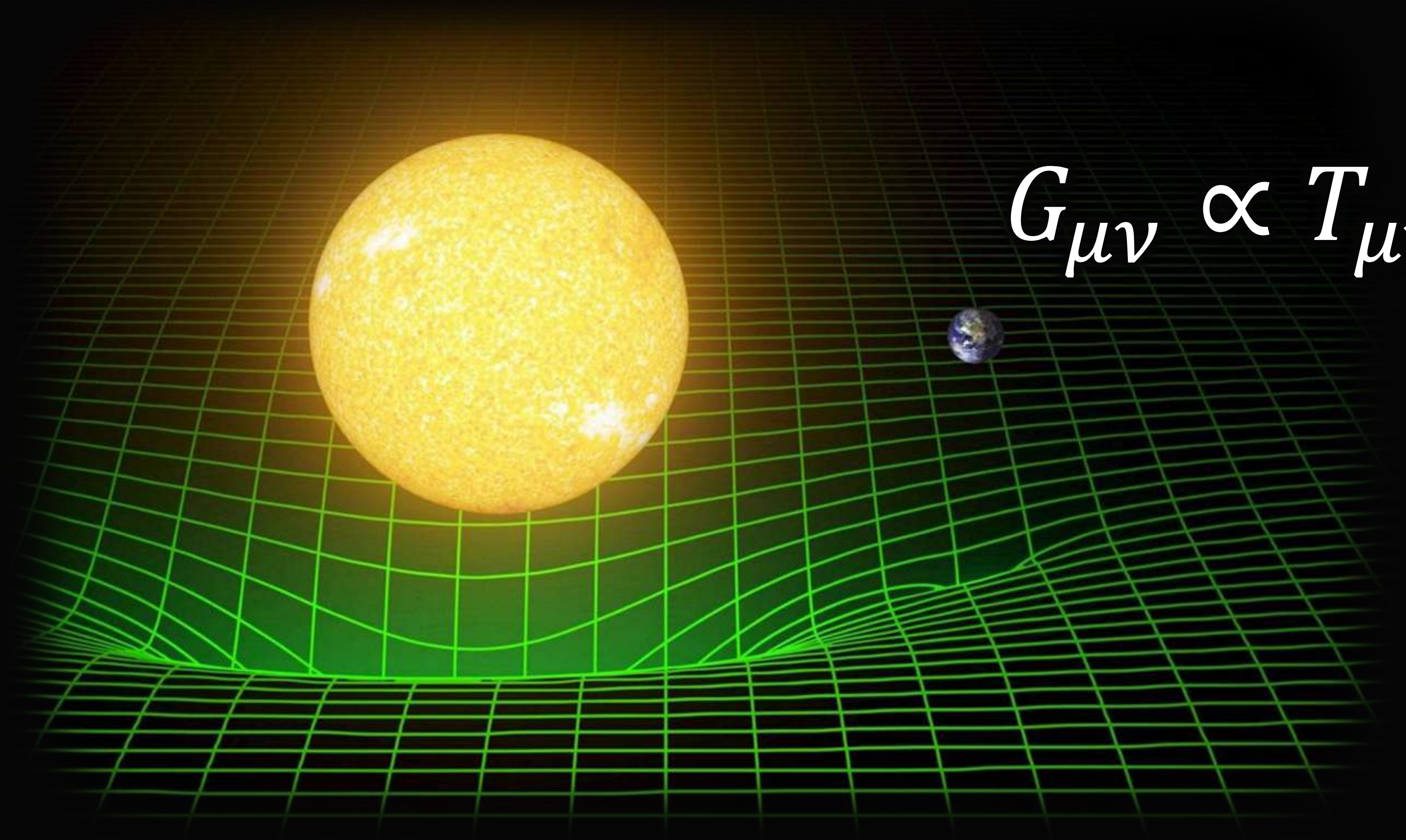
Gravitational Waves FOR DUMMIES*

*MYSELF INCLUDED

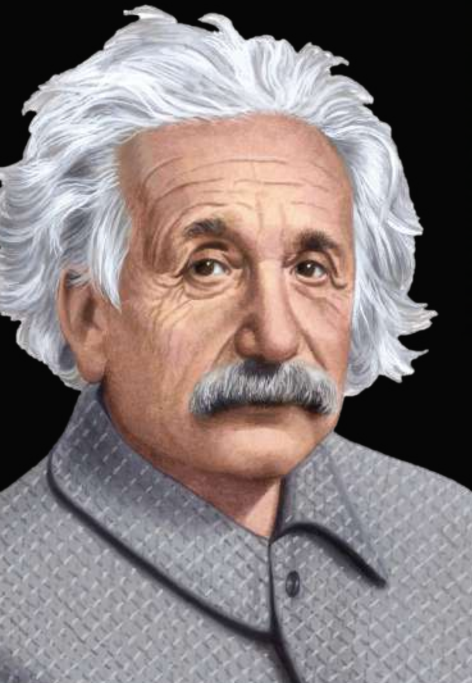
At the beginning there was space…



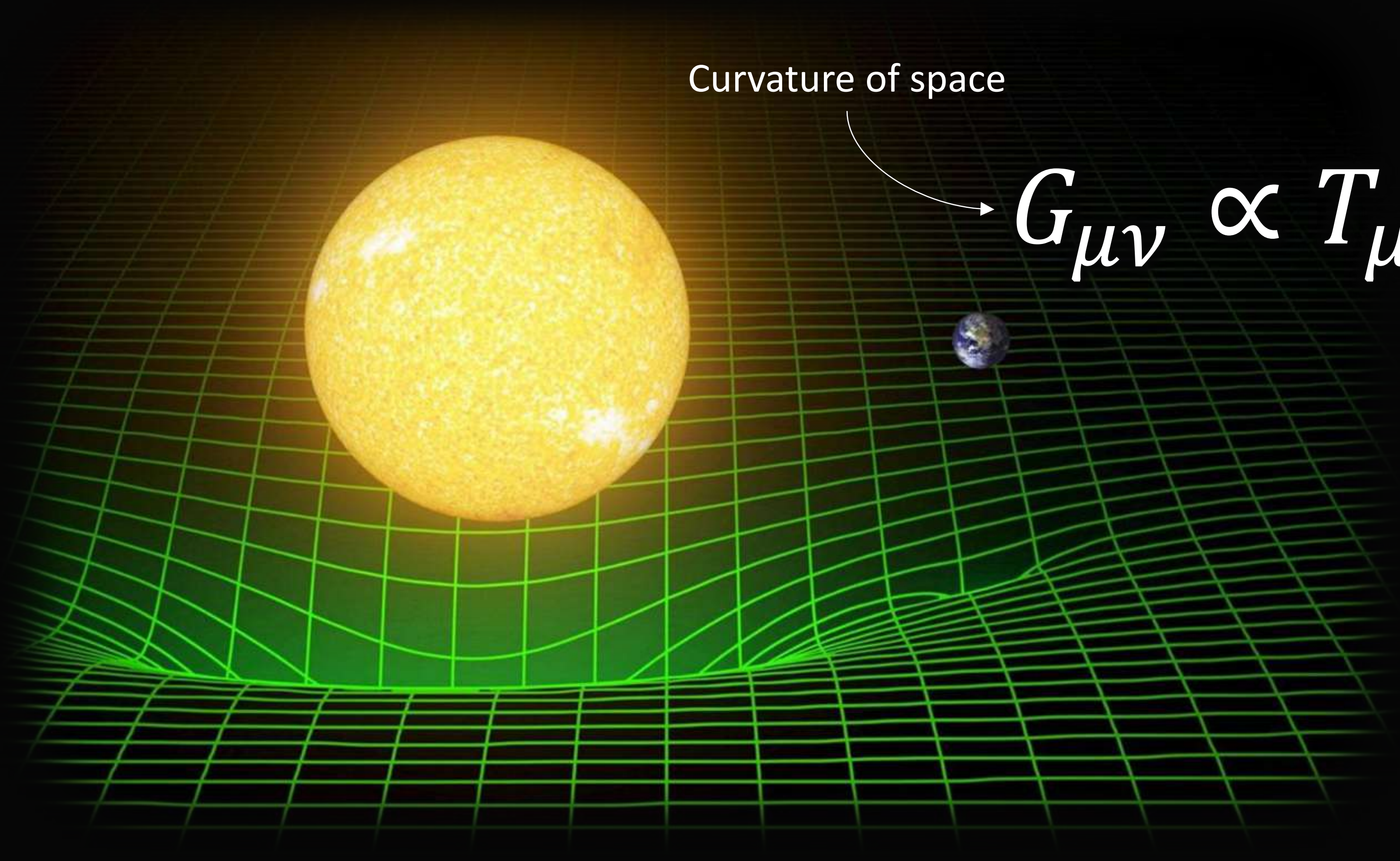
Then tensors appeared…



$$G_{\mu\nu} \propto T_{\mu\nu}$$



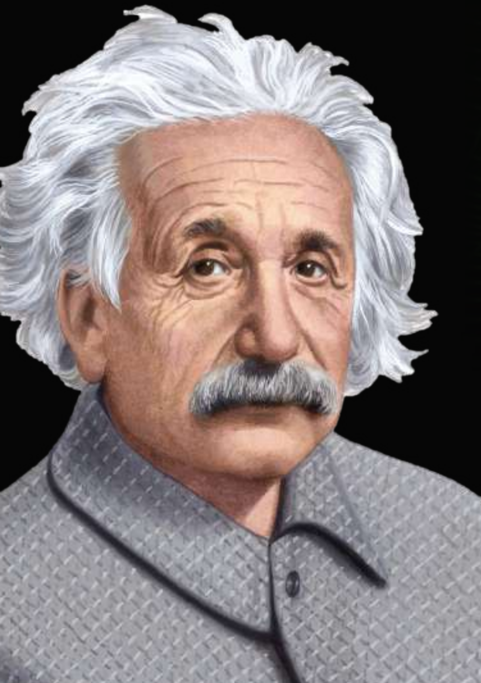
A matter of space



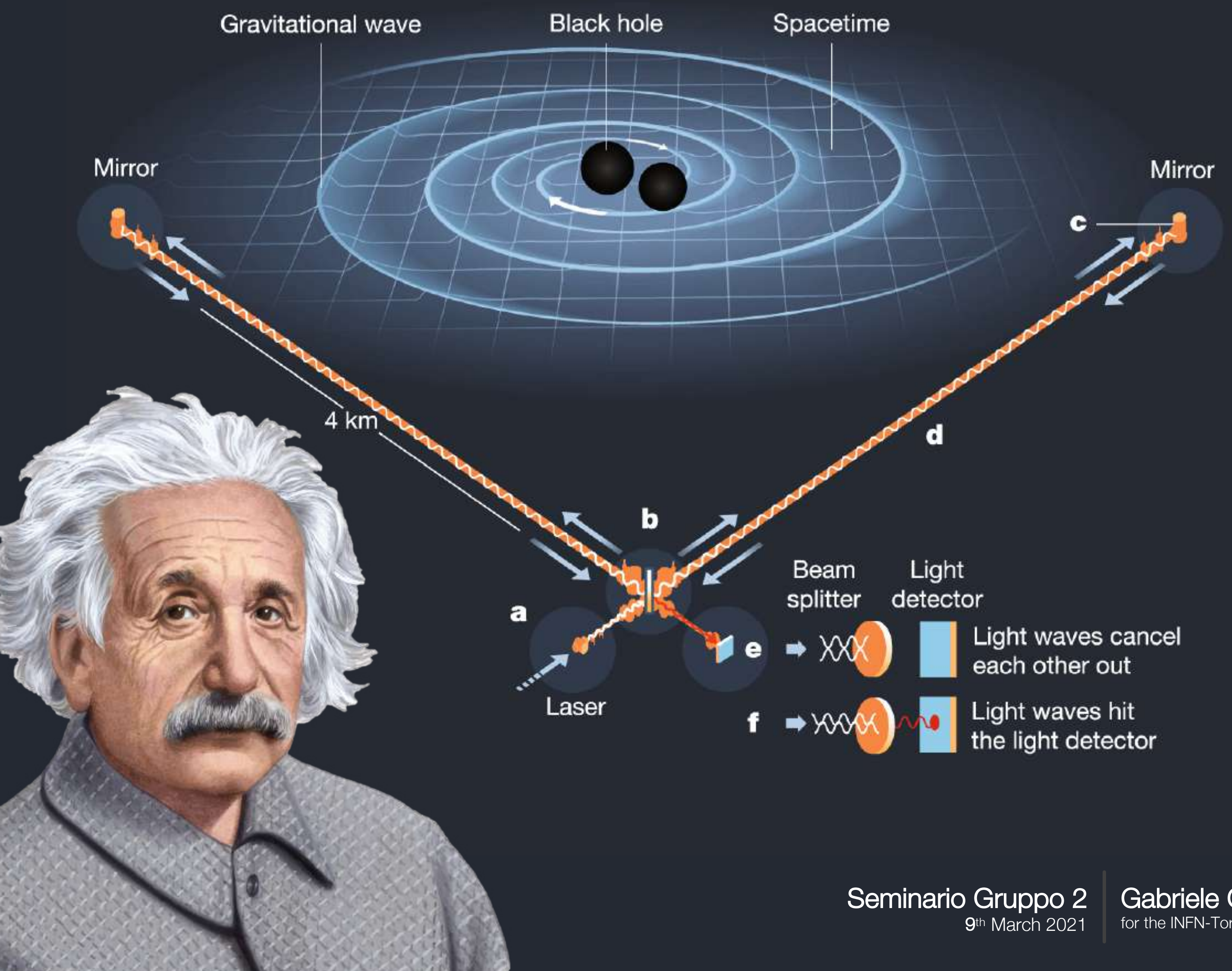
Curvature of space

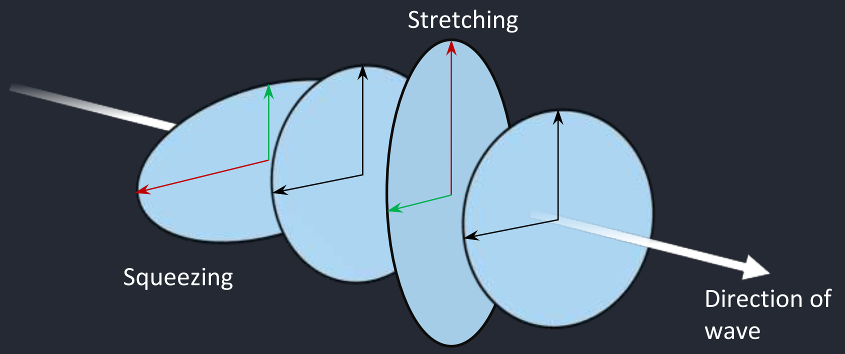
$$G_{\mu\nu} \propto T_{\mu\nu}$$

Energy and mass density



What's a gravitational wave?

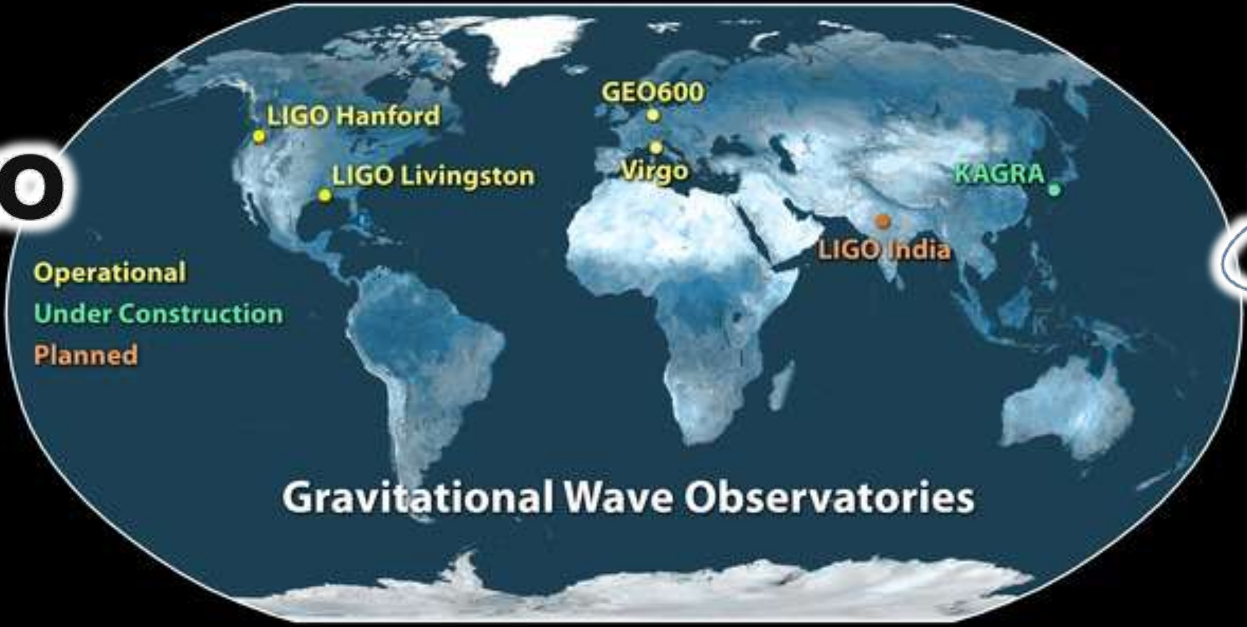




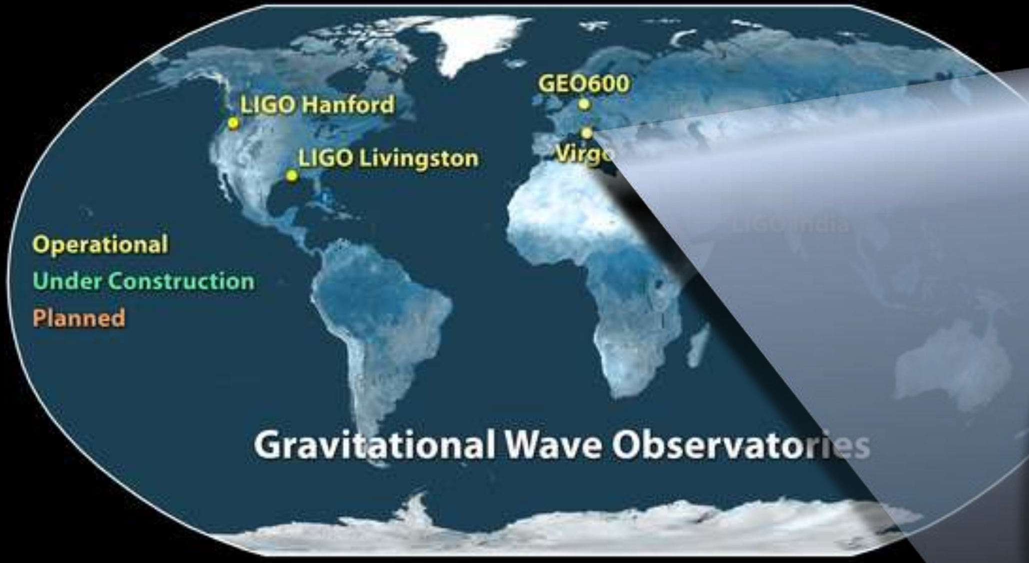
An electromagnetic wave propagates through space through a sinusoidal perturbation of electromagnetic field.

Similarly a gravitational wave propagates through space through a perturbation of the gravitational field, which is the space itself.

A growing network of observatories



A growing network of observatories

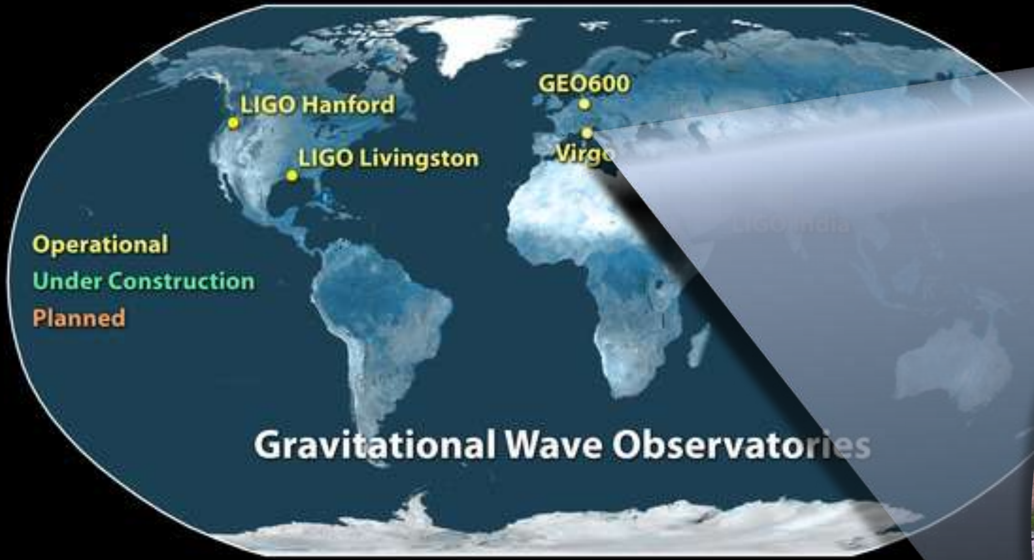


Madonna dell'Acqua
SS12
A12
Colignola
Ripoli
Pisa
Musigliano
Caprona
Zambra
San Giusto
PSA
Riglione
La Vettola
Titignano
Navacchio
VIA DEL FOSSO VECCHIO
VIA AURELIA SUD
VIA NUGOLAIO
VIRGO
Coltano
US Army Camp Darby Italy
VIA ARNACCIO
AUTOSTRADA AZZURRA
Vicarello
Guasticce

Virgo is one of the gravitational interferometers installed around the globe.

It is part of a collaboration with LIGO and KAGRA.

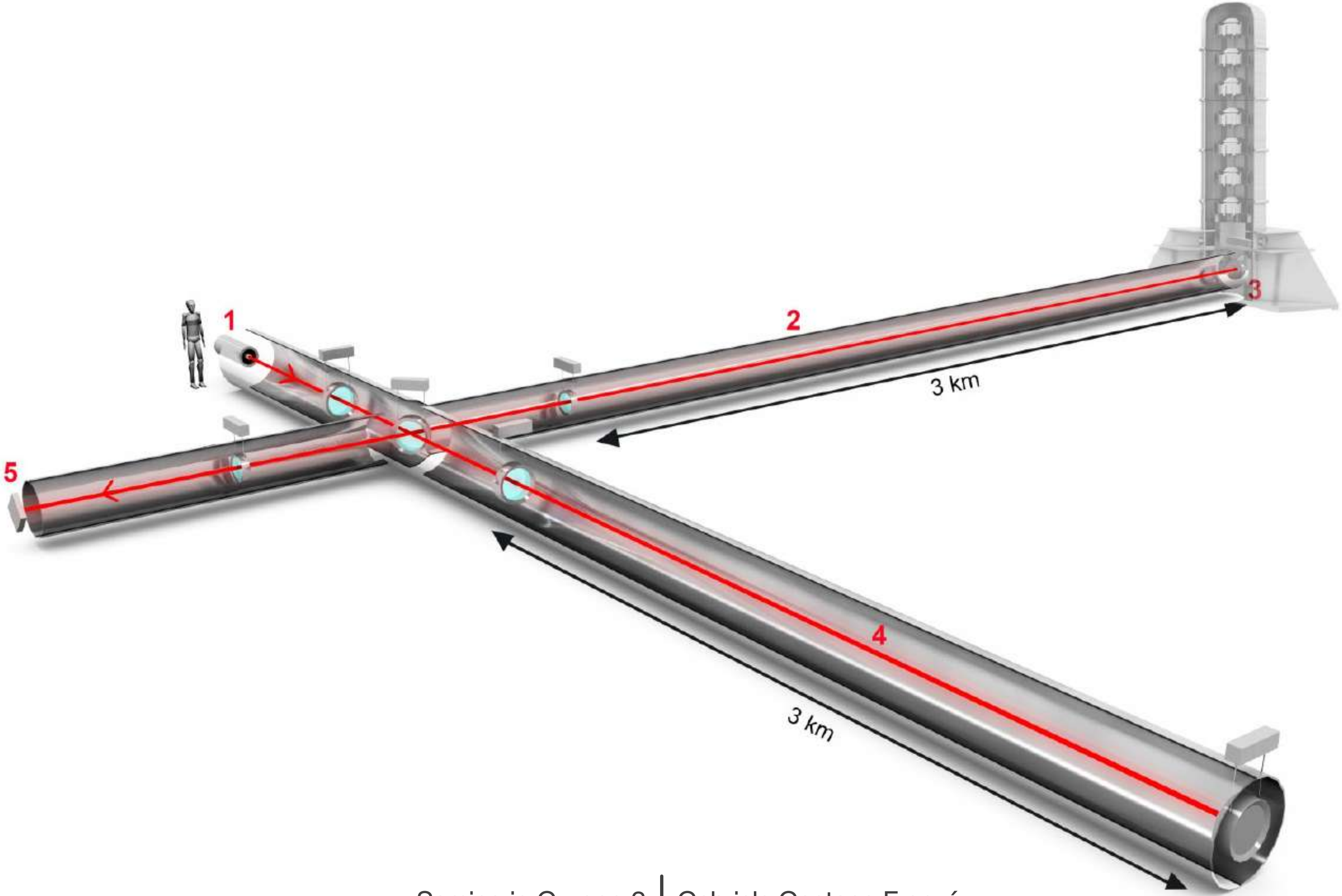
A growing network of observatories



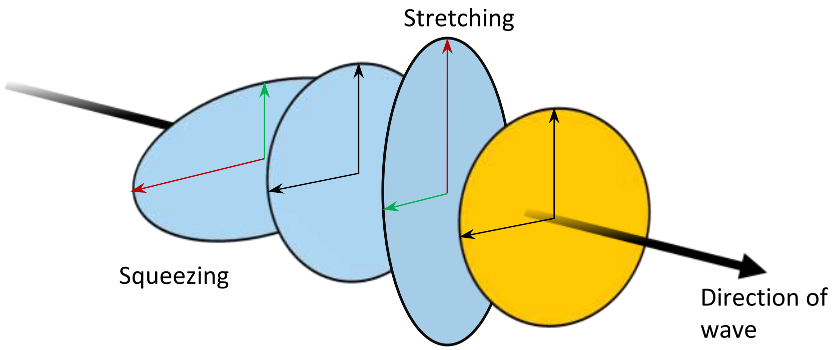
Virgo is one of the gravitational interferometers installed around the globe.

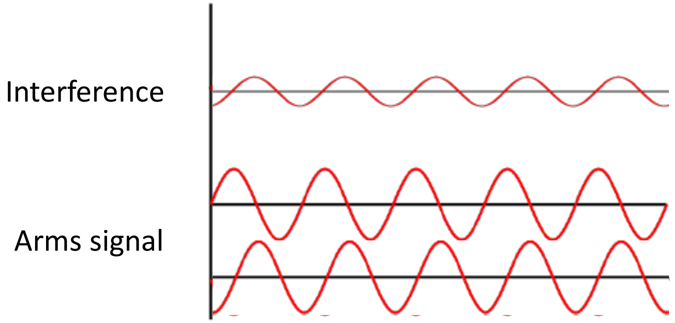
It is part of a collaboration with LIGO and KAGRA.

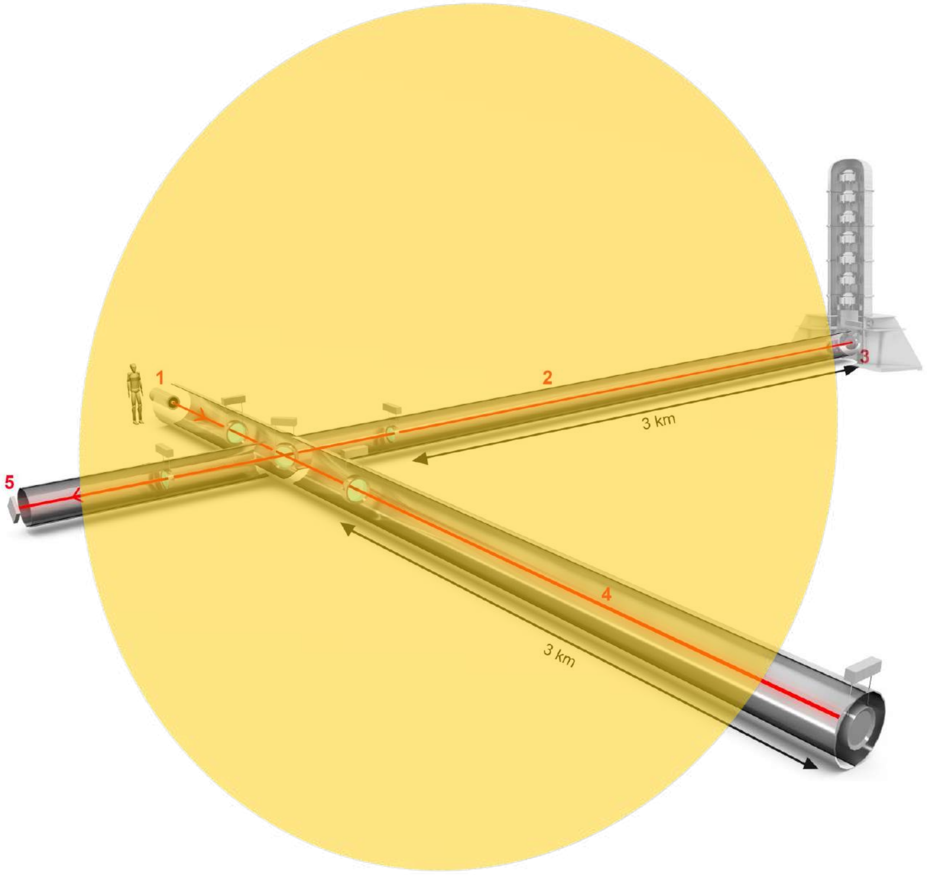
How is a gravitational interferometer made?



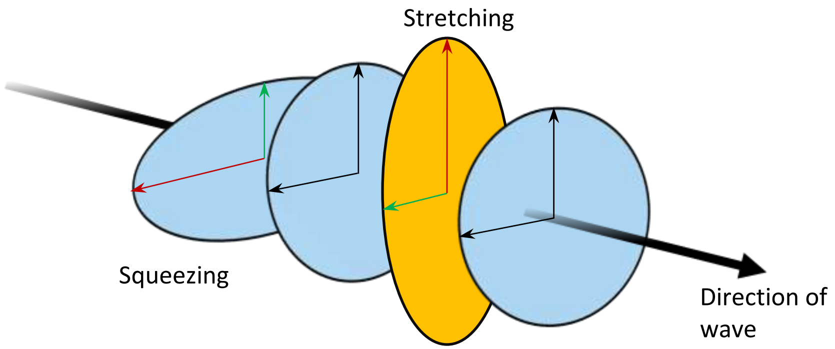
How does it work?

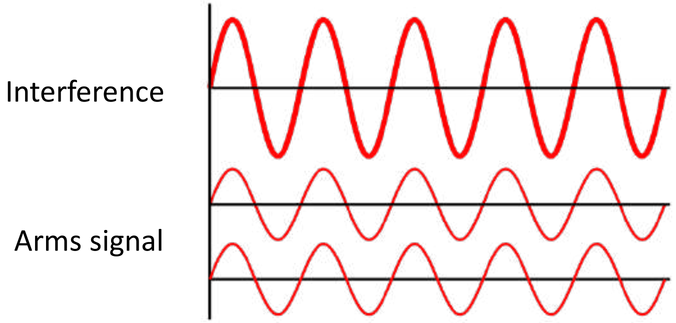


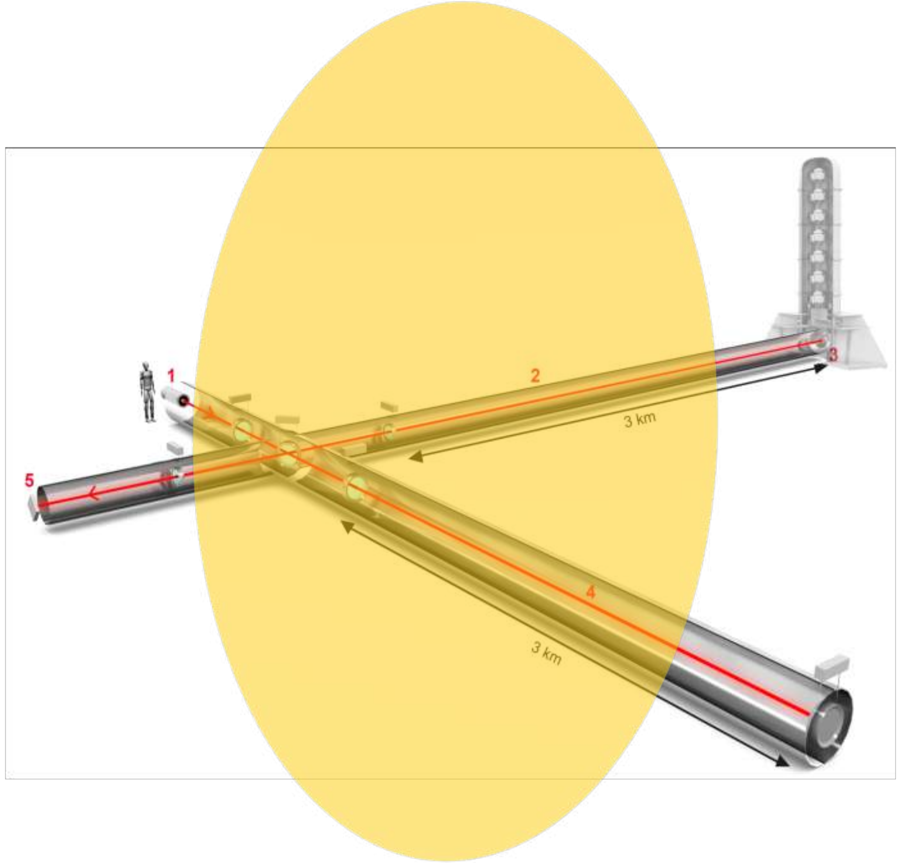




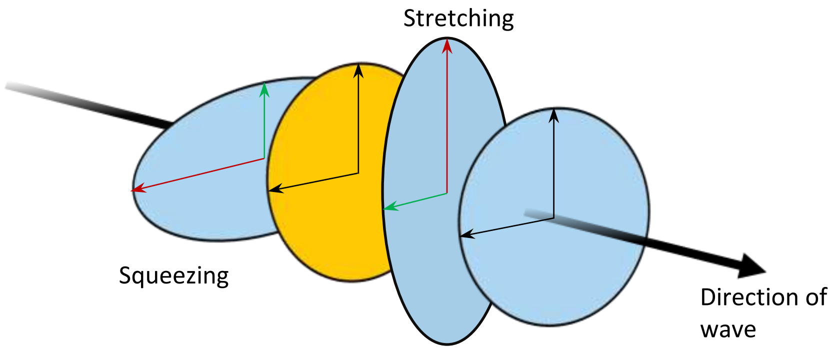
How does it work?

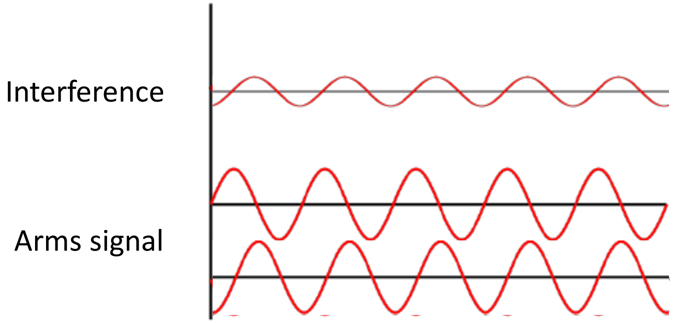


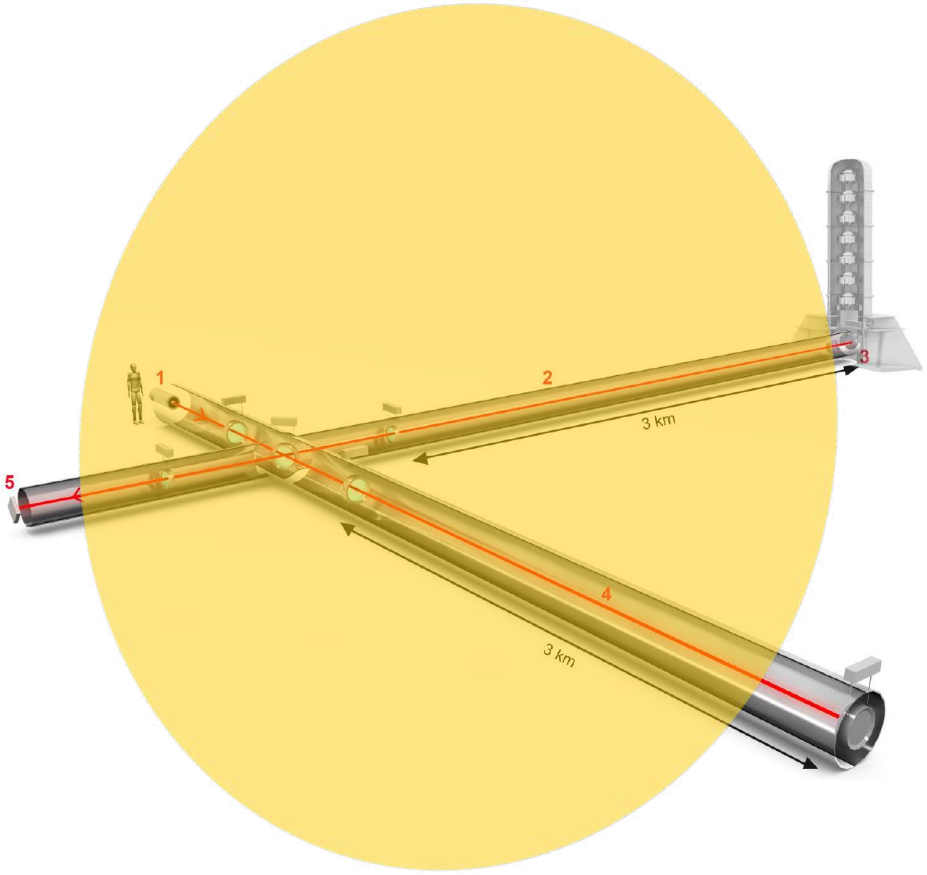




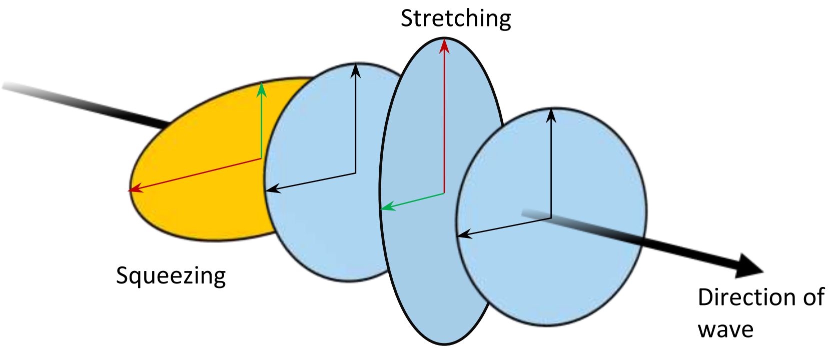
How does it work?

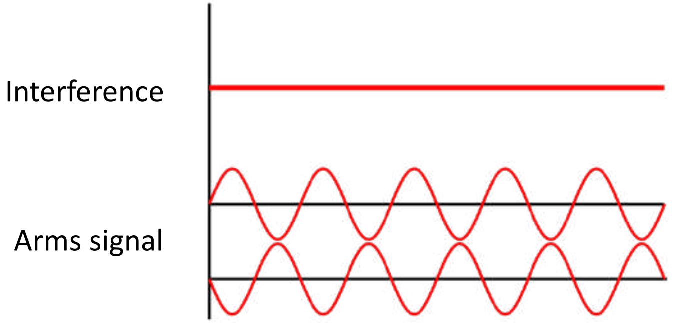


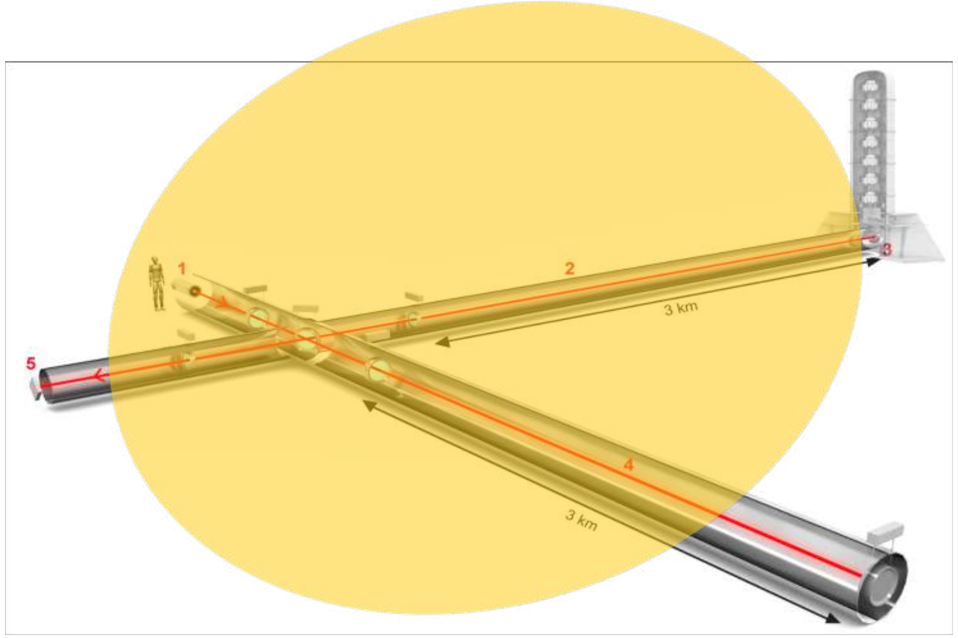




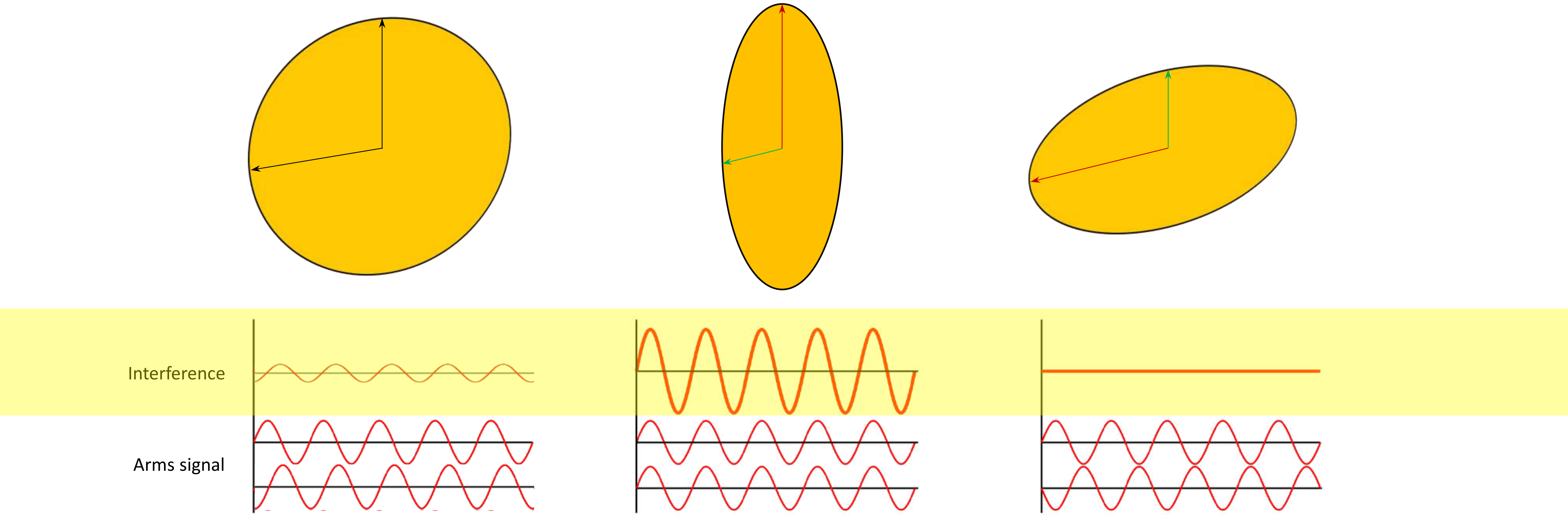
How does it work?







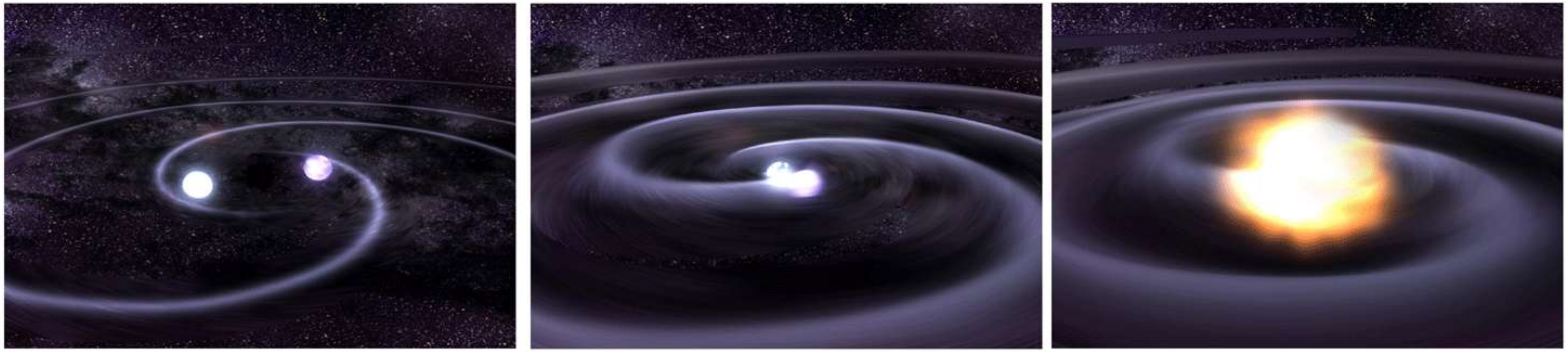
How does it work?



Record the interference amplitude and you are ready to go… almost

$$interference\ amplitude = H(t) = strain$$

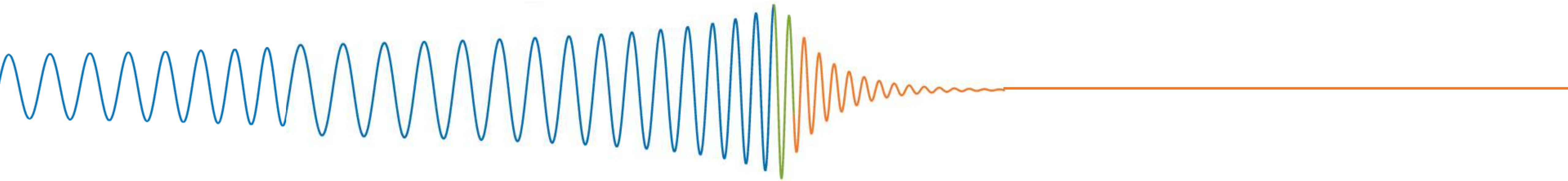
What would you expect?



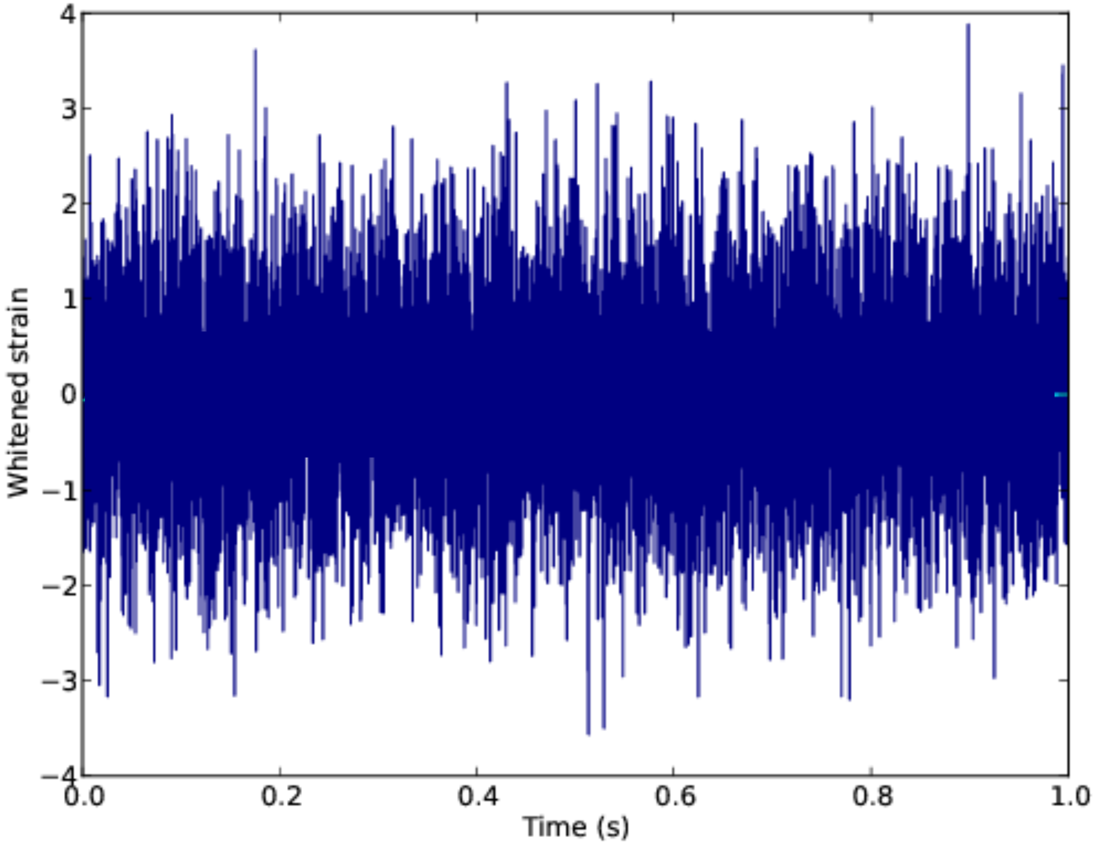
Inspiral

Merger

Ringdown



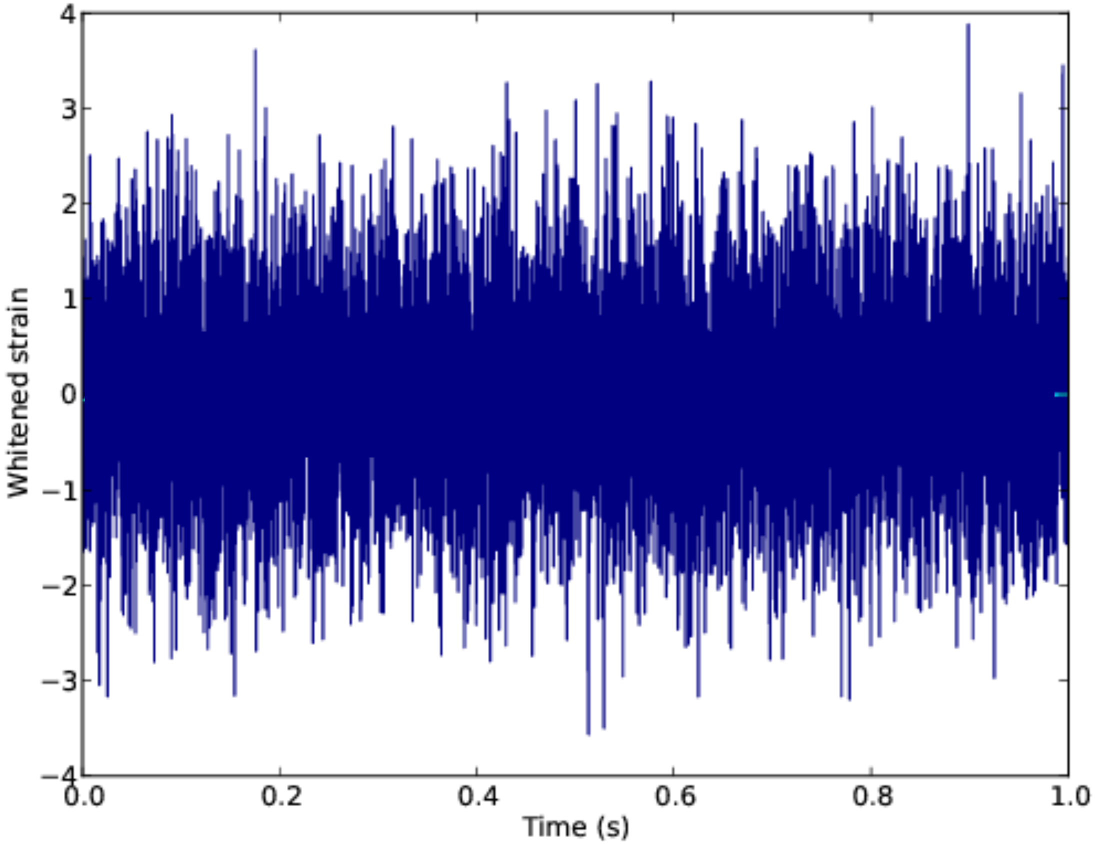
Get the measured spectrum…



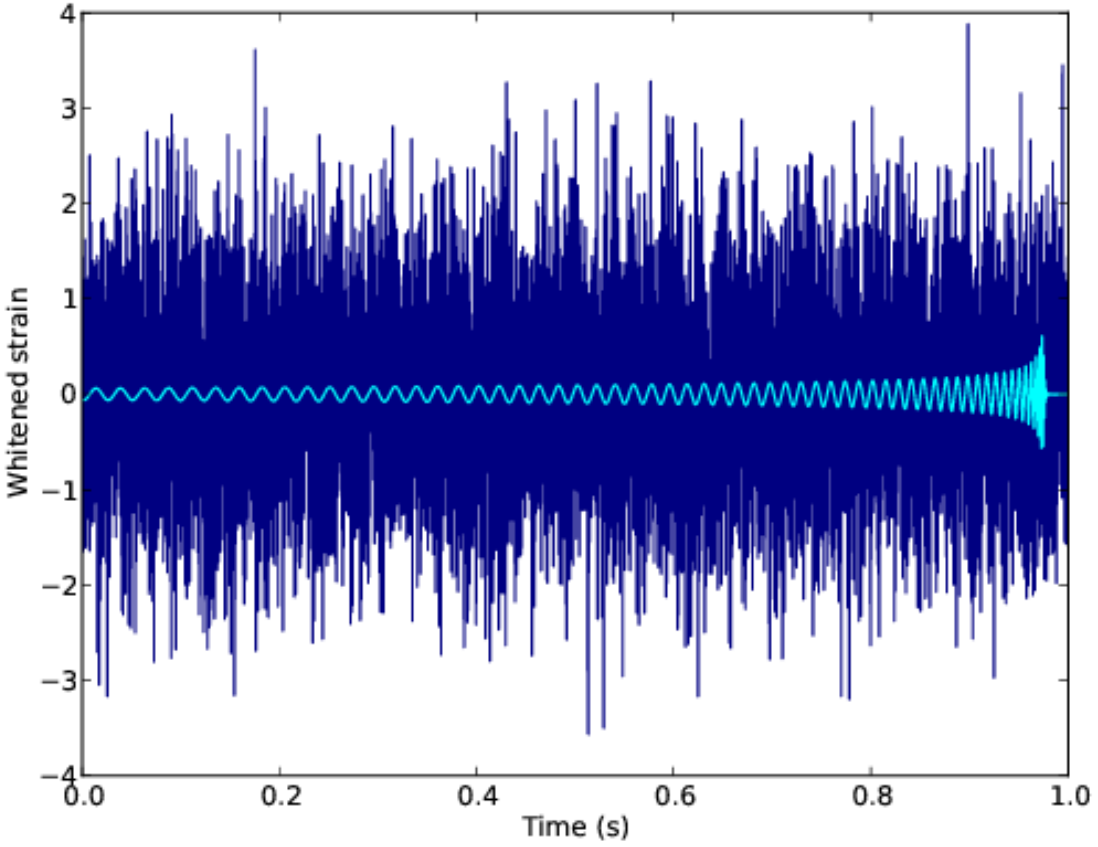
Is the amplitude enough?

And the noise?

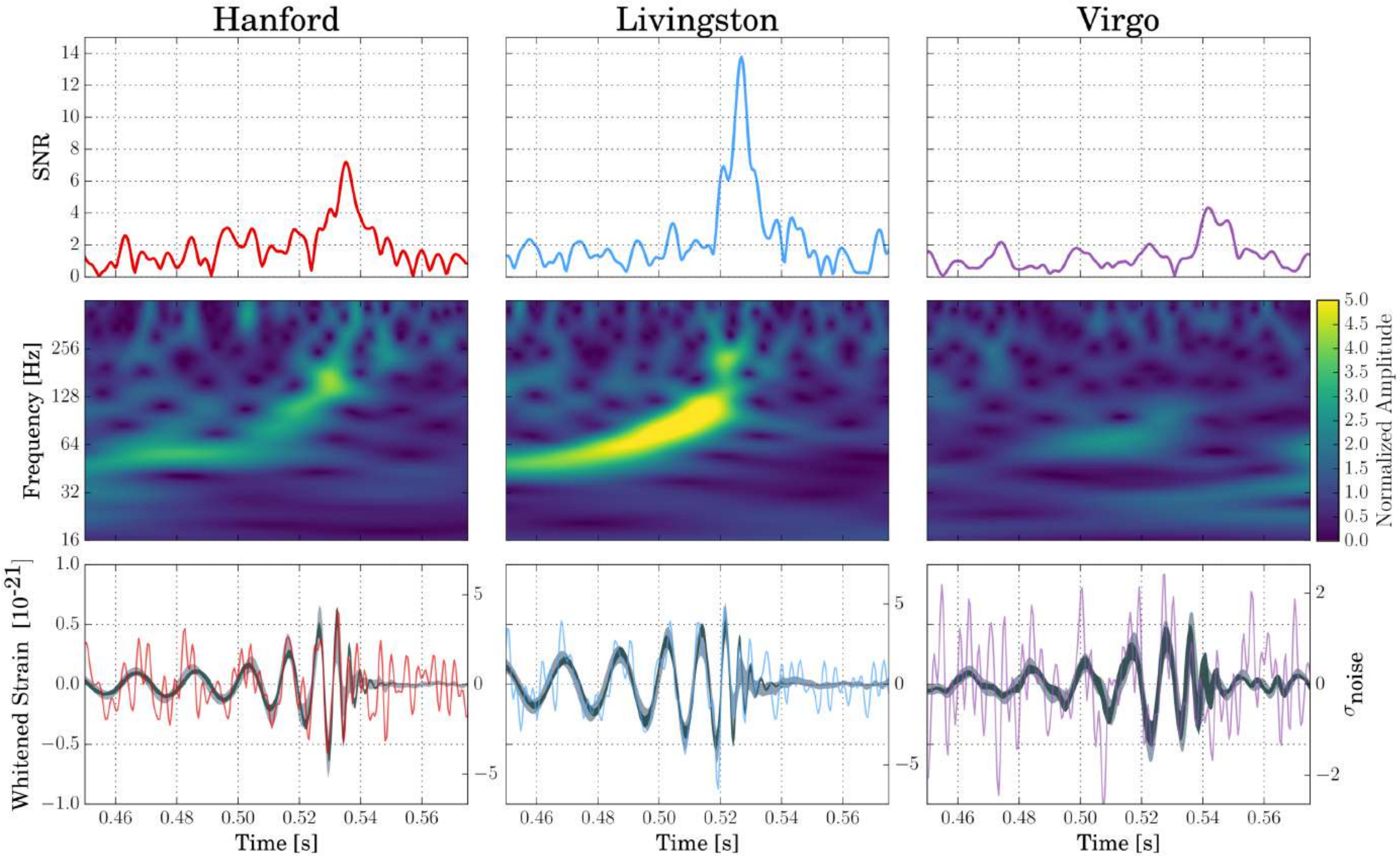
Remove thermal noise, doppler, distortion…



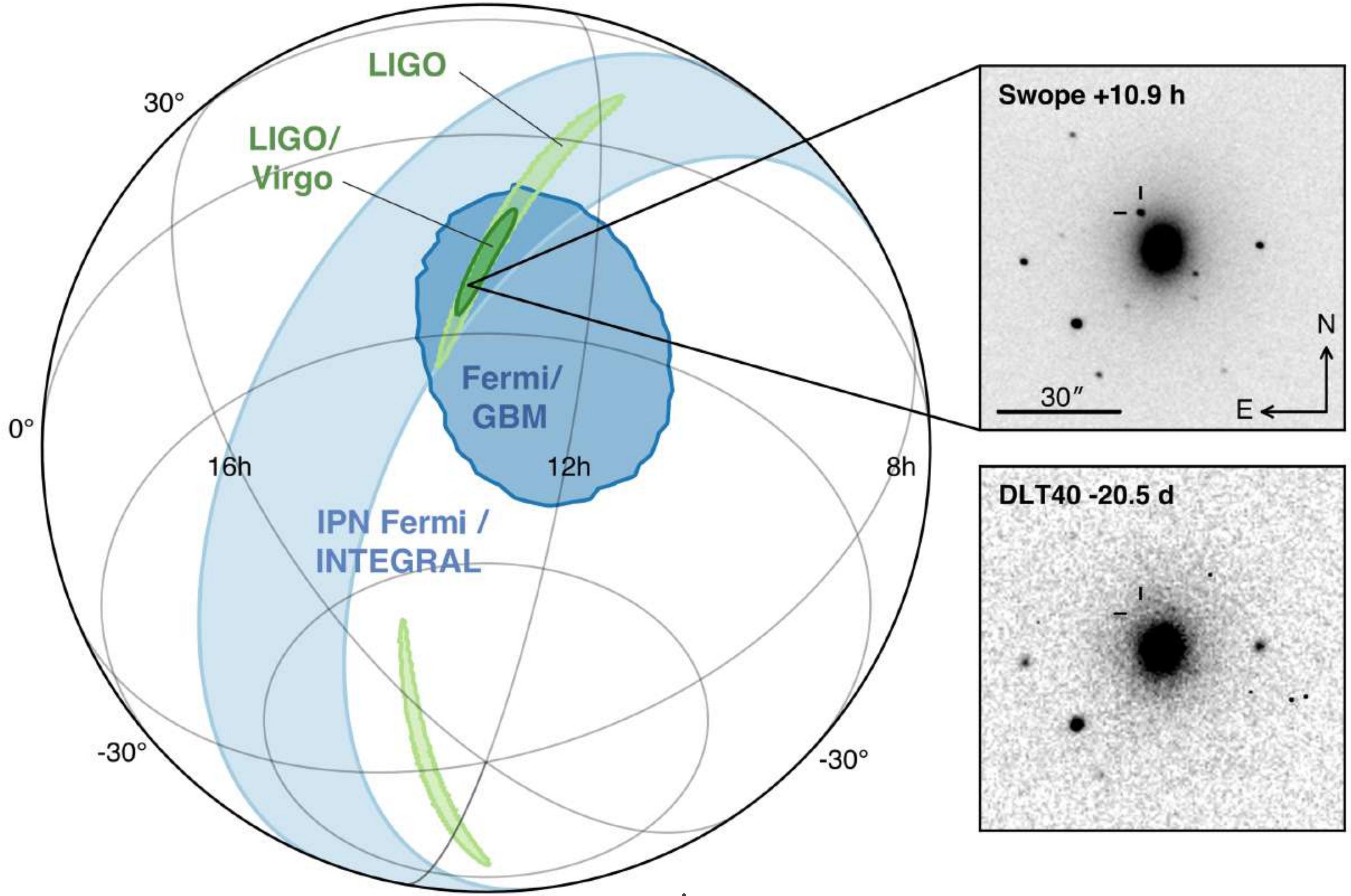
Remove thermal noise, doppler, distortion…



Triangulate…



The power of unity



"Multi-messenger Observations of a Binary Neutron Star Merger"
B. P. Abbott et al. 2017 ApJL 848 L12 doi:10.3847/2041-8213/aa91c9

COMPUTING ASSETS

OR HOW TO GET AWAY WITH A COMPUTING MODEL

Three assets

Storage

Software

Computing

Four questions

- Where to put your raw data? Who needs to read it?
- How to build and distribute your software?
- How to process and analyze data? Who should do that?
- Where to put analysis outputs?

A crucial distinction

Backend

How do things work and how to manage them.
End users should marginally know of its existance.
Should be handled by professional IT crew.

Coffee!

The only thing you can take as granted…

Frontend

The scientists’ user experience baseline.
End users should be involved in decisions regarding this aspect.
Might be delegated to scientists.

Data distribution (1)

Backend

The first component of the storage backend is the catalog.

It should allow for:

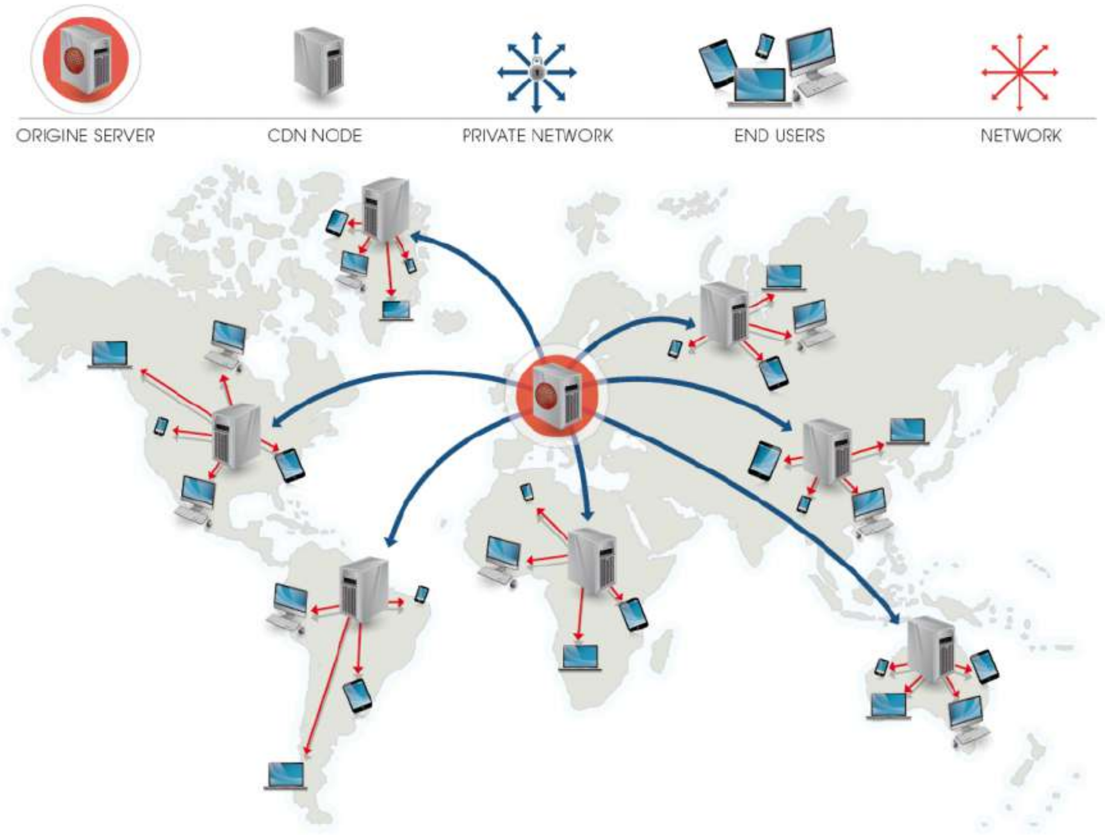
- Federation of storage resources:
  In order to enable aggregation of storage resources provided by multiple institutions
- Data bookkeeping:
  To provide a catalog of replicas and to provide metadata-based queries for data selection
- Handling of data replication:
  To automatically create data replicas and to avoid data losses even in case of accidents

It's a crucial component which typically relies on databases and a bunch of services.

Your data is your value, don't be afraid of spending the right money to put in place a solid distributed deployment (hardware and software).

Data distribution (1)

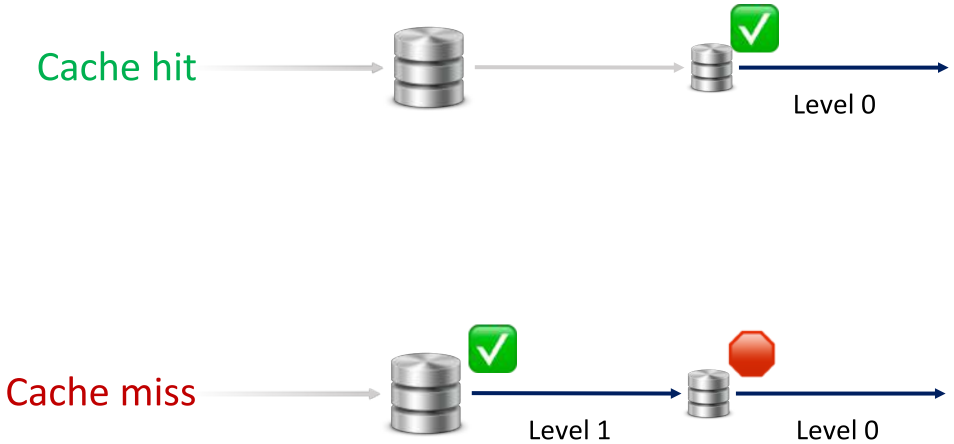
Backend



Rely on existing (or build your own) Content Delivery Network.

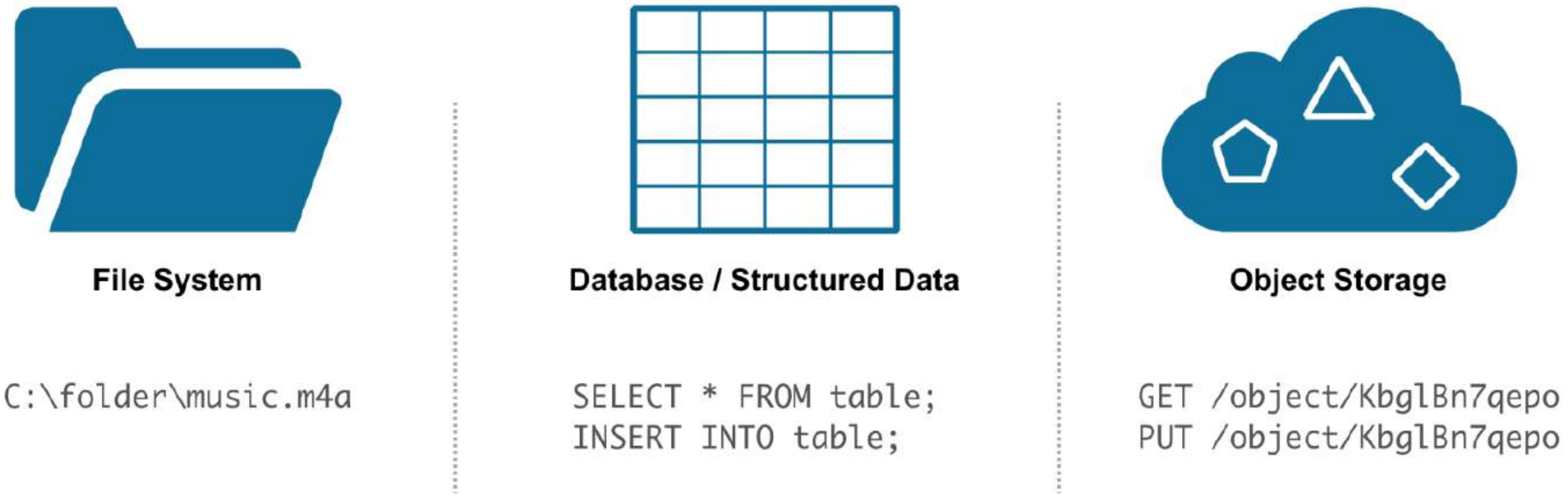
A CDN is a cache hierarchy which automagically diffuses data as soon as they are requested.

Technicalwise, your experimental data is not dissimilar from a Netflix movie!



Data distribution (2)

Frontend



File System

```
C:\folder\music.m4a
```

Database / Structured Data

```
SELECT * FROM table;
INSERT INTO table;
```

Object Storage

```
GET /object/KbglBn7qepo
PUT /object/KbglBn7qepo
```

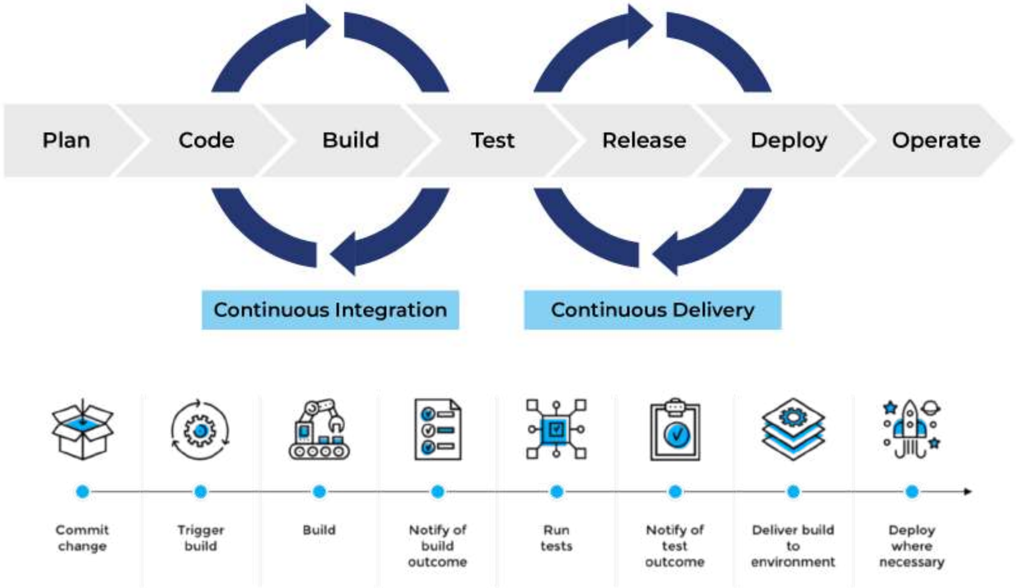
Do the users expect a folder tree, a database-like or a “each data item has a specific URL” data access structure?

What’s the UI they are more familiar with?

Can we afford multiple access patterns on the same backend?

Software distribution (1)

Backend



Software distribution is merely the last link in the chain.

Always write tests for your code, in order to avoid regressions.

Use a modern version control tool (e.g. git).

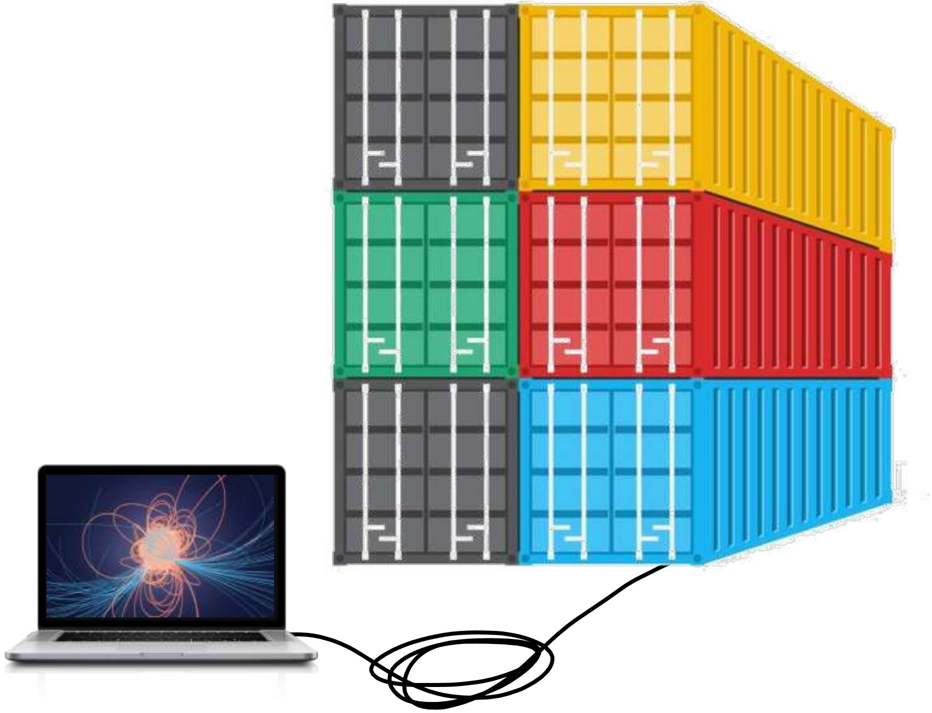
Put in place a solid CI and CD (Continuous Integration and Continuous Delivery) pattern to automatically build, test and distribute your software.

Major corps are using the same tools, don’t be afraid of learning from them!

Software storage is a one-way system: make CD automatically publish new releases into you framework of choice and distribute them across the subscribers (aka workstations and computing nodes).

Software distribution (2)

Frontend



You software should be as atomic as possible.

Pack dependencies with your software: give the users a ready-made environment to run the task by simply plugging in the data.

Containers and virtual environments are your allies.

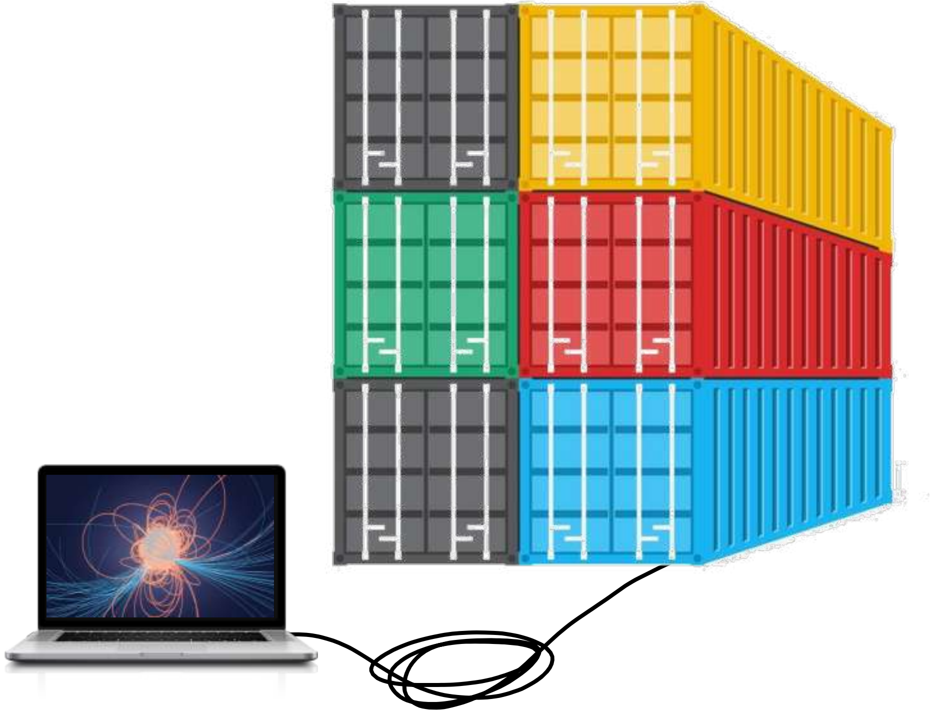
Allow your users to adopt the same behavior on their own workstations or on remote computing resources.

Enable effort-less local testing.

Provide an understandable POSIX structure with version hierarchy.

Software distribution (3)

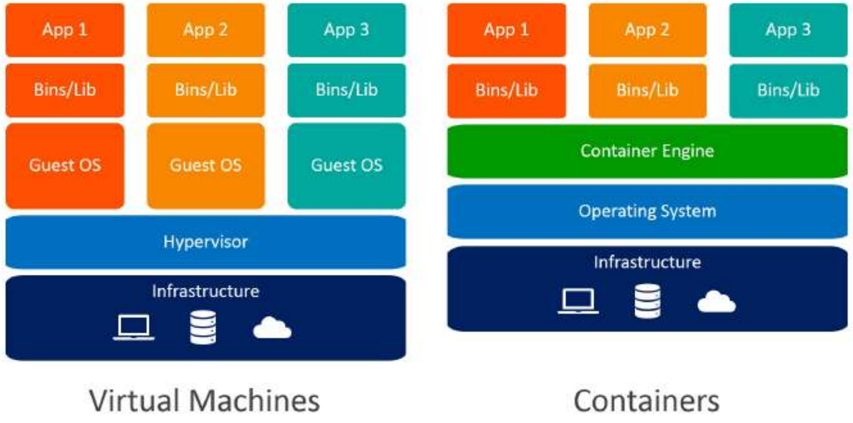
Frontend



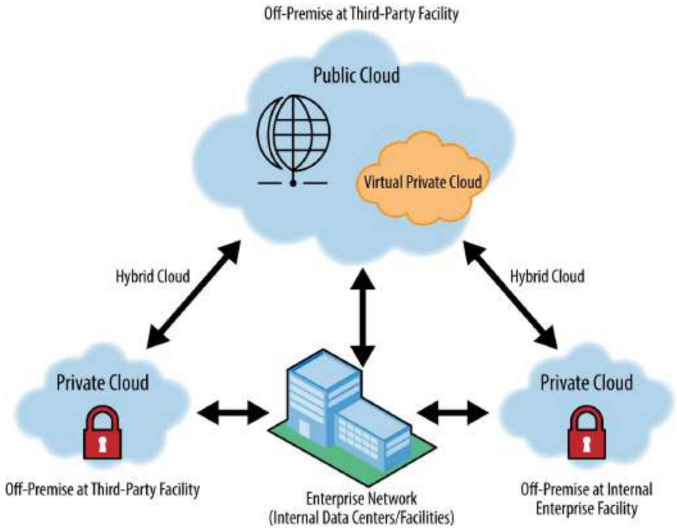
A container is a kernel-less virtual machine, which is run on the host kernel as a process.

It's quite similar to a BSD Jail (or chroot if you are familiar with it).

Since the image doesn't contain the OS, the size is merely that of the data and executables shipped within.



Computing infrastructure (1)



Backend

A computing infrastructure must be capable of scaling out easily.

Build your computing backend with the idea of scaling on public clouds in case of need.

Streamline process of integrating new computing resources.

HTCondor is a de facto standard born in the scientific community and adopted by corporations and national agencies. It is complex to deploy, but highly versatile.

If your software is distributed as containers, consider using a container orchestrator (i.e. kubernetes) to do the work. Container orchestrators are great at running container. You don't say?

Think of a way to EASILY declare inter-job dependencies and sequential workflows.

HTCondor High Throughput Computing

kubernetes

Computing infrastructure (2)

Backend

Is that a service people uses all the time?

Yes

No

Stay home

Go cloud

Computing infrastructure (3)

Frontend

Allow users to test locally during debugging and to test the runtime before bulk submission.

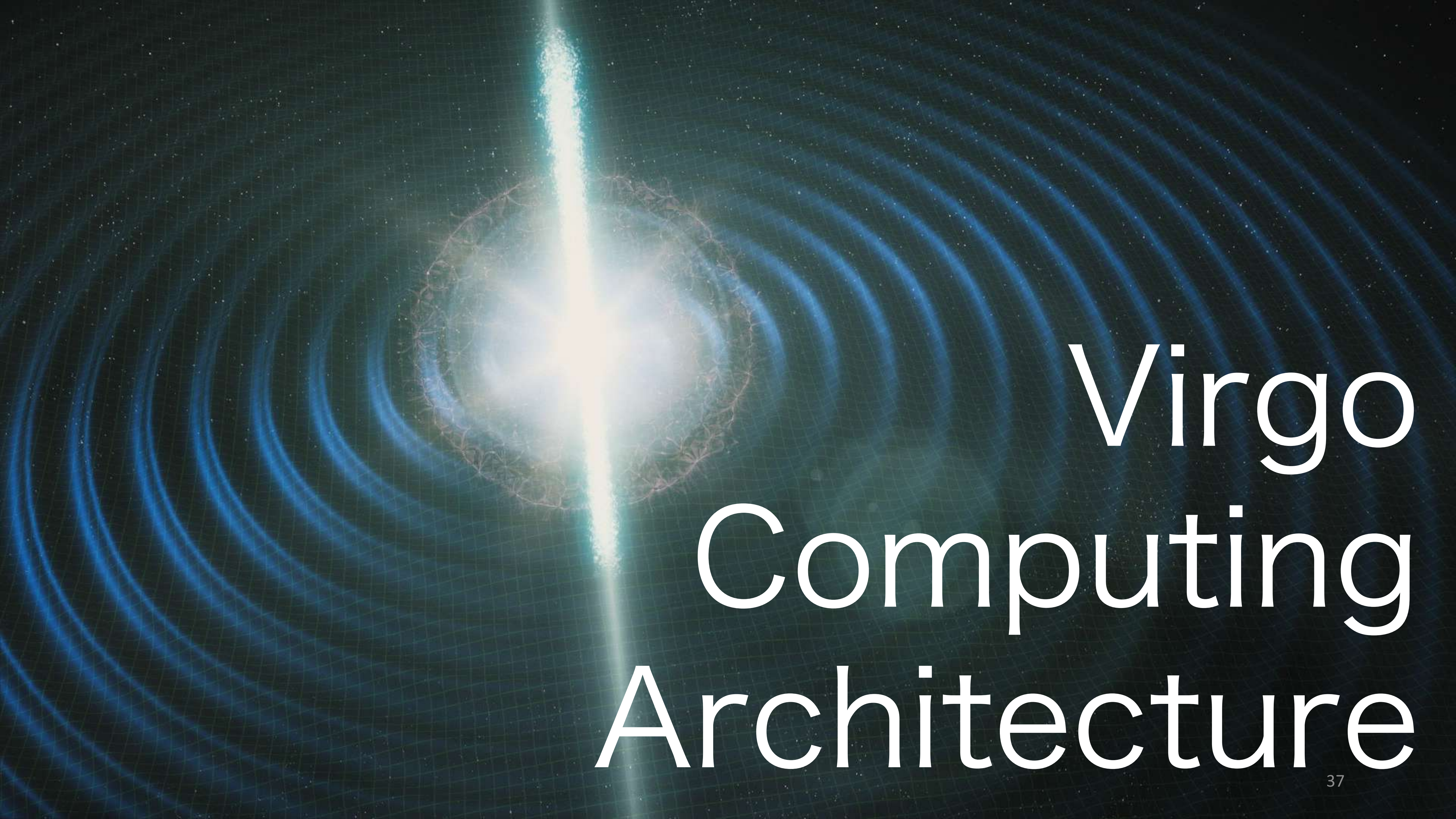
Allow users to seamlessly transition from their workstation to the computing infrastructure.

Provide a dashboard (web or CLI, doesn’t matter) to easily check job status, recover failed ones and, more in general, inspect their computations.

Define a pattern to easily handle the computations’ output, by publishing it on the collaboration resources or sending it to the submitter private storage.

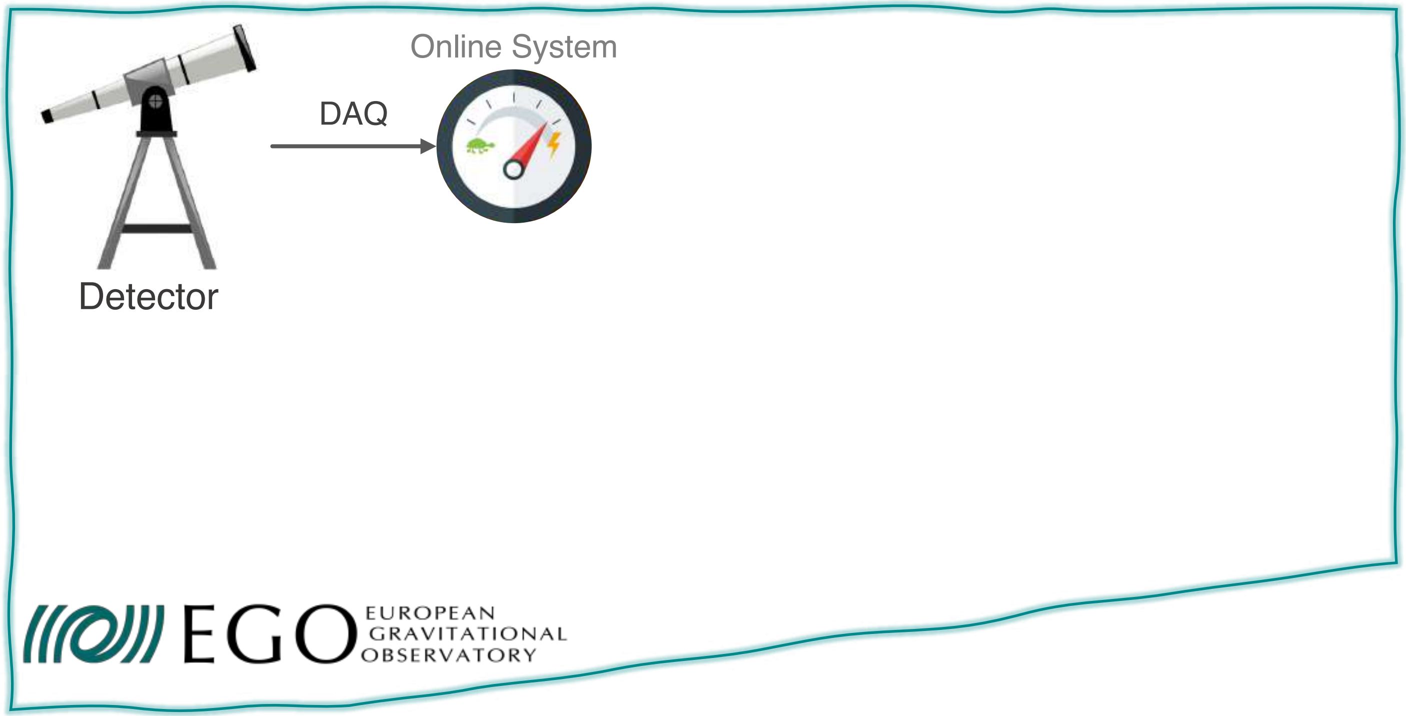
Train your people to fully exploit the resources you put together, trainign is crucial, not just a best effort practice.

Make sure users can access a reliable and accessible documentation.



Virgo Computing Architecture

Data sources, distribution and replication



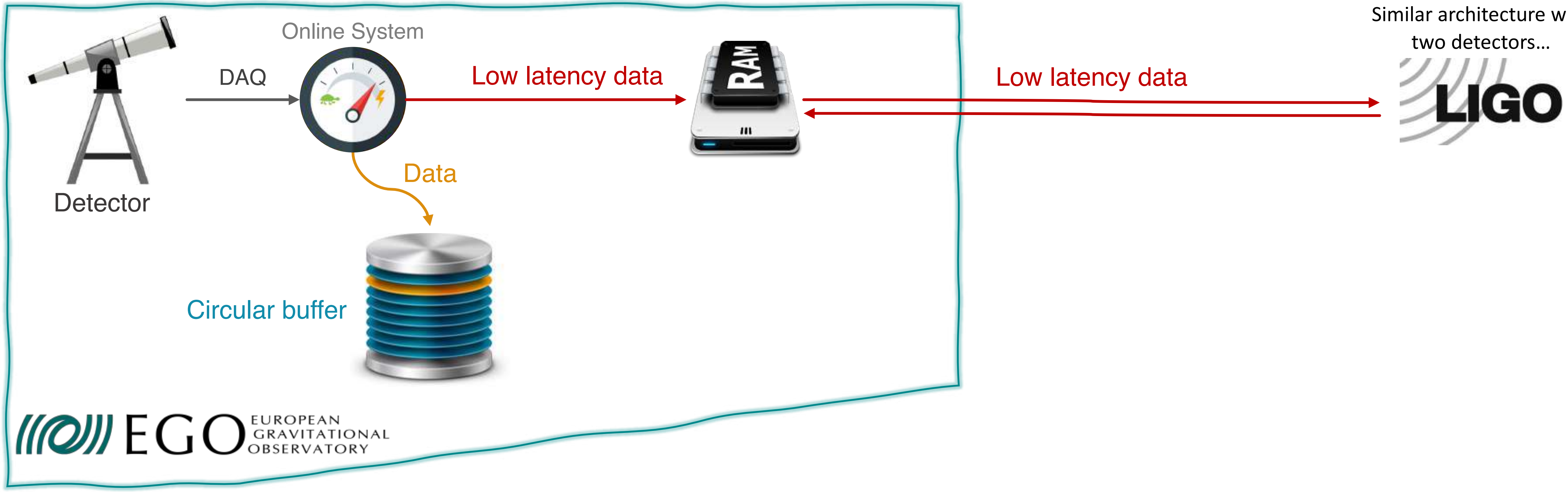
Online System

DAQ

Detector

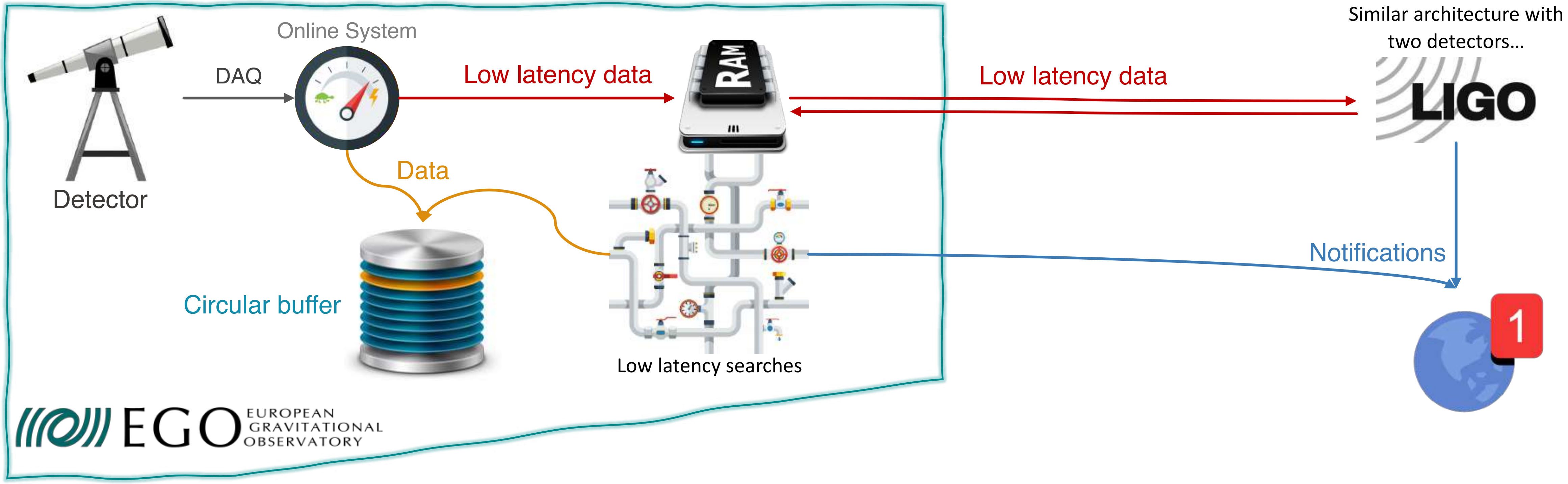
EGO EUROPEAN GRAVITATIONAL OBSERVATORY

Data sources, distribution and replication

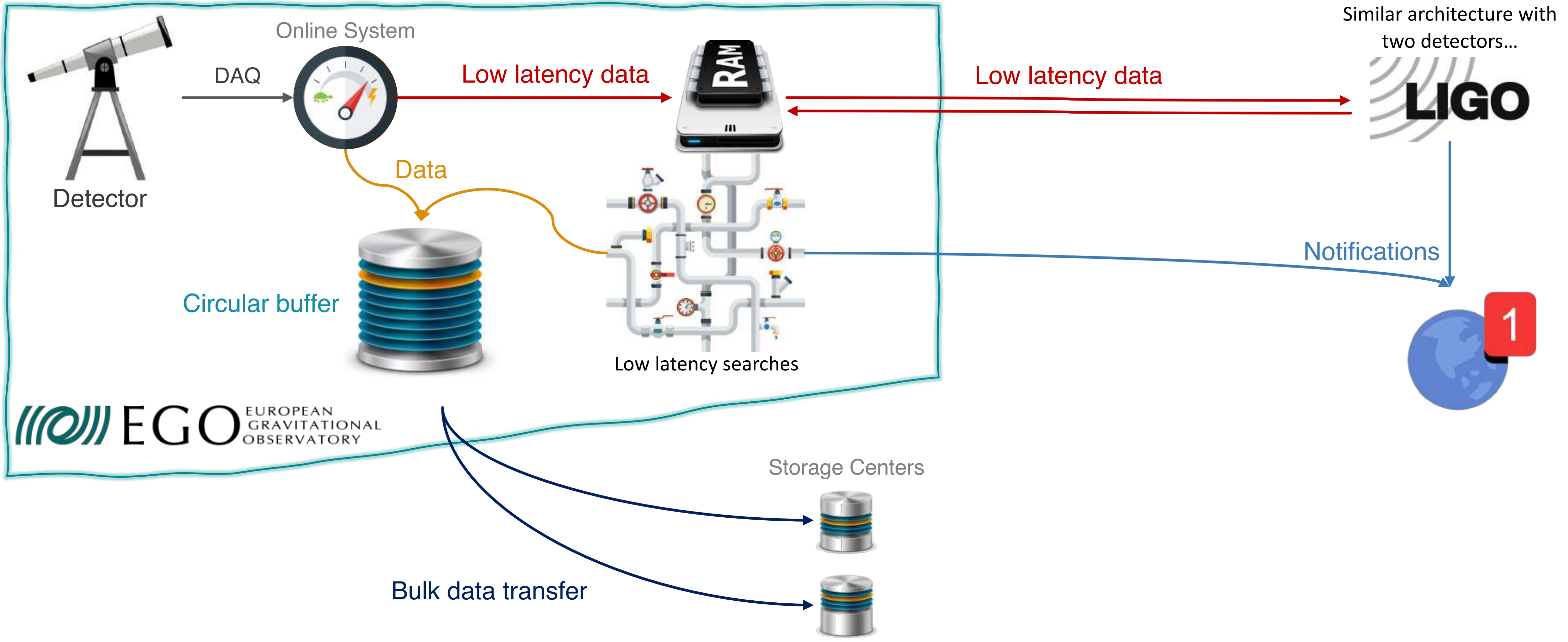


Similar architecture with two detectors…

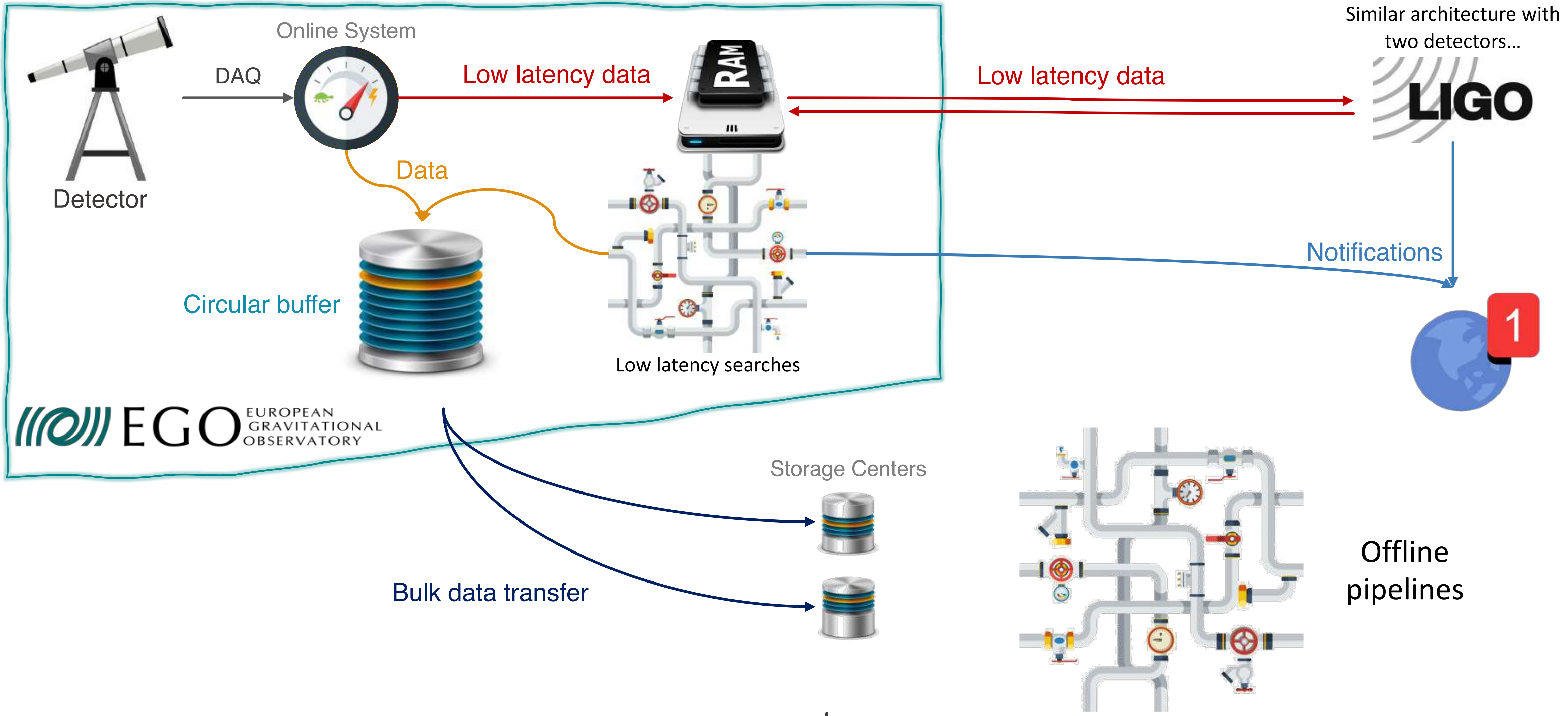
Data sources, distribution and replication



Data sources, distribution and replication



Data sources, distribution and replication



Online System

DAQ

Detector

Low latency data

Data

Circular buffer

Low latency searches

Similar architecture with two detectors…

LIGO

Low latency data

Notifications

EGO EUROPEAN GRAVITATIONAL OBSERVATORY

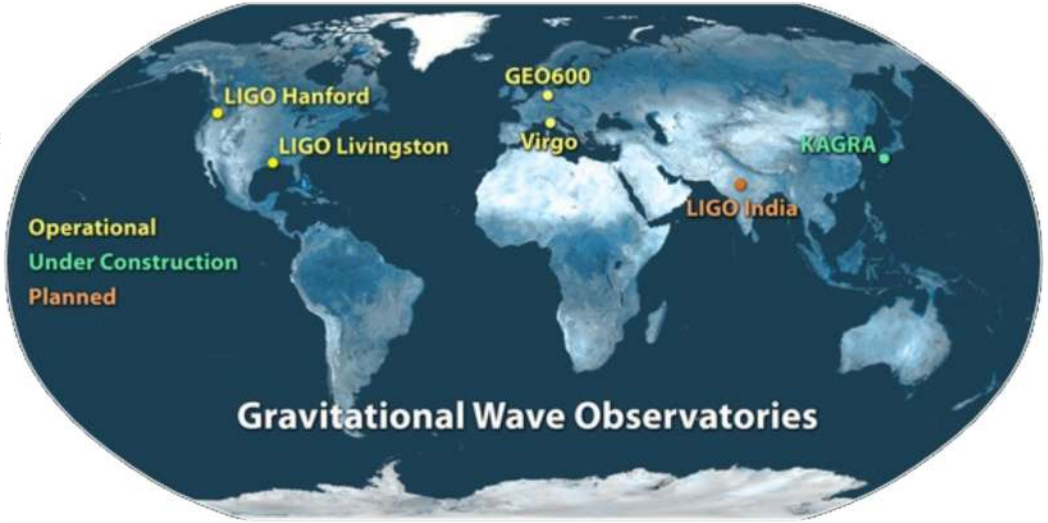
Storage Centers

Bulk data transfer

Offline pipelines

LVK and IGWN

LVK = LIGO, Virgo, KAGRA



The detectors are spread around the world and a centralized infrastructure is impossible
but
a symmetric infrastructure is highly recommended

LVK and IGWN

IGWN = International Gravitational-Wave observatory Network

Currently the best effort to:

- Solve computing infrastructure asymmetries
- Define standards for new requirements
- Help transition legacy solutions to such standards

Three assets (recap)

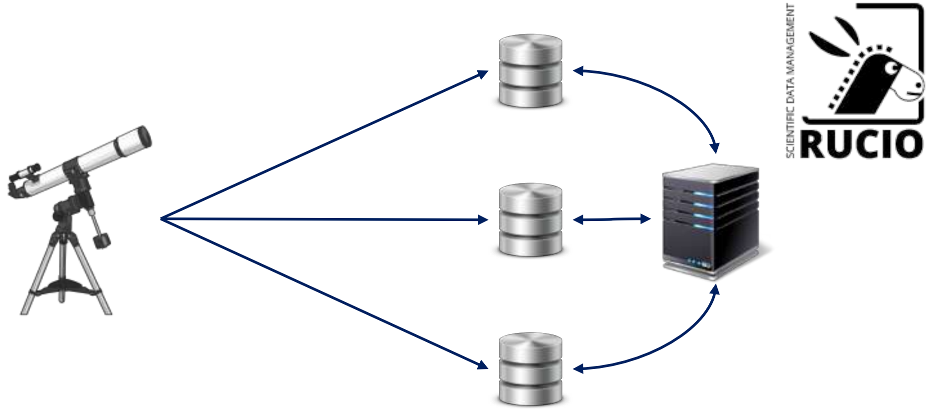
Storage

Software

Computing

Storage (1)

Backend



Rucio is the tool of choice.

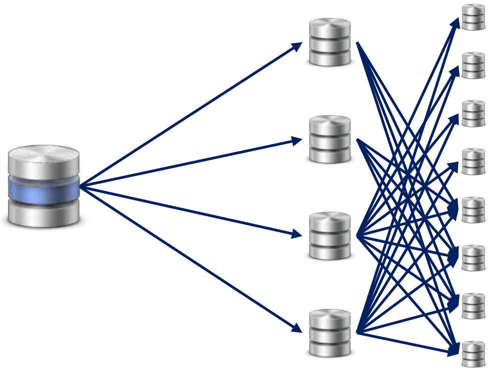
It was born for the ATLAS collaboration at LHC, but is getting adopted by lots of collaborations.

It federates storage endpoints and handles replicas and metadata.

It is compatible with GridFTP, Storm, S3, WebDav, XRootD and many more and supports x509, token, userpass and other authentication mechanisms.

Storage (2)

Backend



XRootD

A CDN based on StashCache/xCache which automatically diffuses the most requested files “agnostically”.

Frontend

CernVM-FS

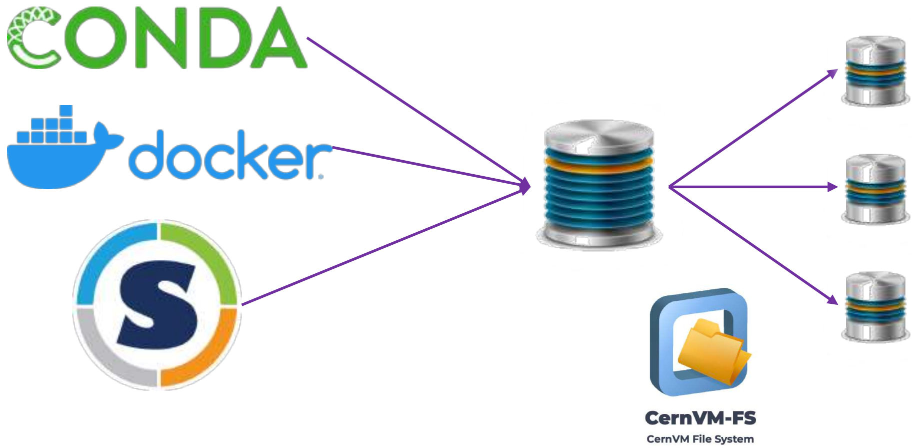
CernVM File System

CVMFS-based POSIX representation of data.

Users are used to POSIX and habits are difficult to change!

Software

Backend



A CVMFS deployment on which developers and CI/CD workflows can automatically push new software, packed as virtual environments, containers or plain executables.

Frontend

CernVM-FS

CernVM File System

CVMFS-based POSIX representation of data.

Users are used to POSIX and habits are difficult to change!

Computing

Backend

A federated HTCondor computing pool spanning multiple computing centers around the world.

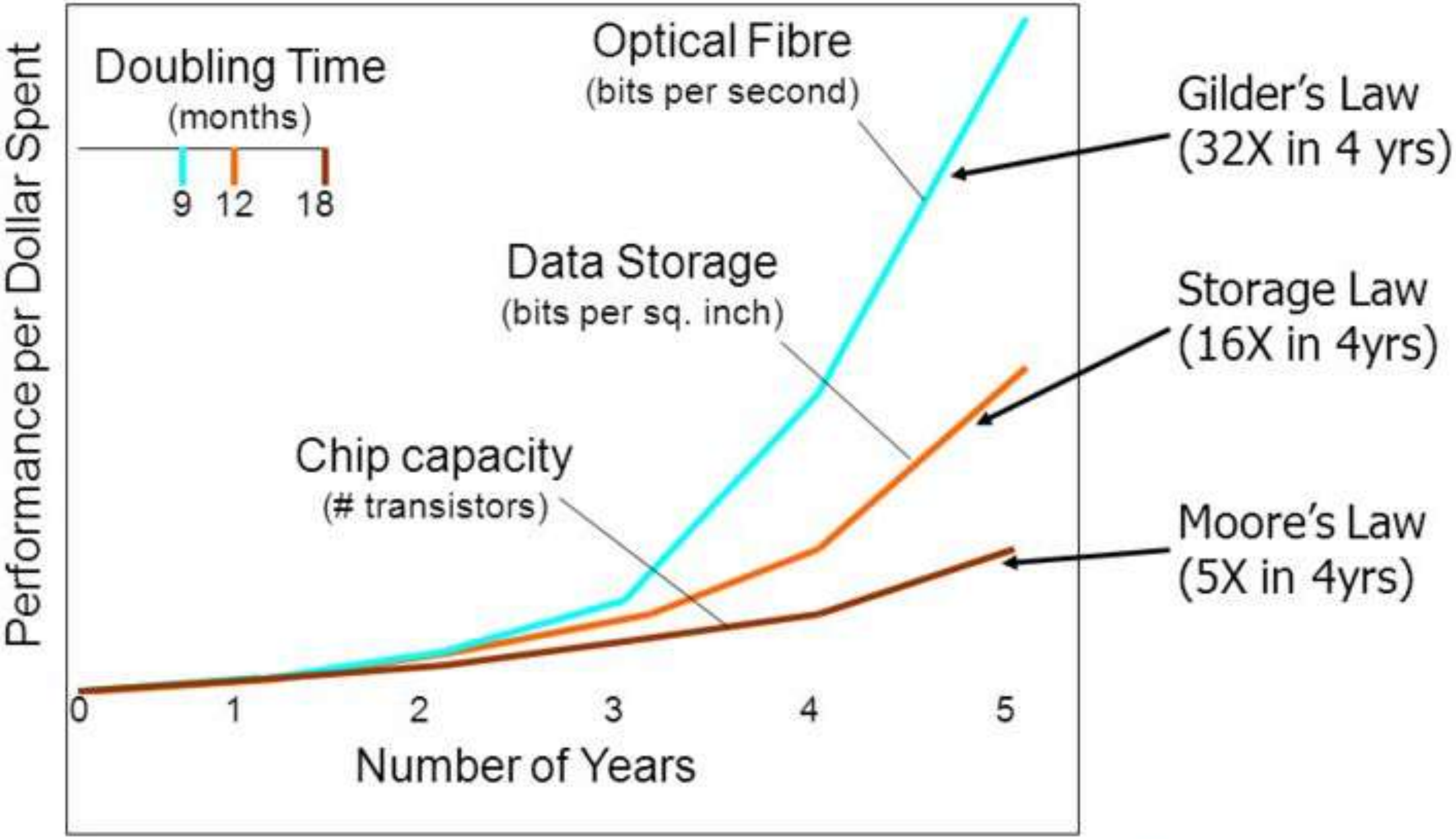
Frontend

The HTCondor CLI and its automatic output backpropagation, all installed on UI machines displaced in the collaboration CCs.

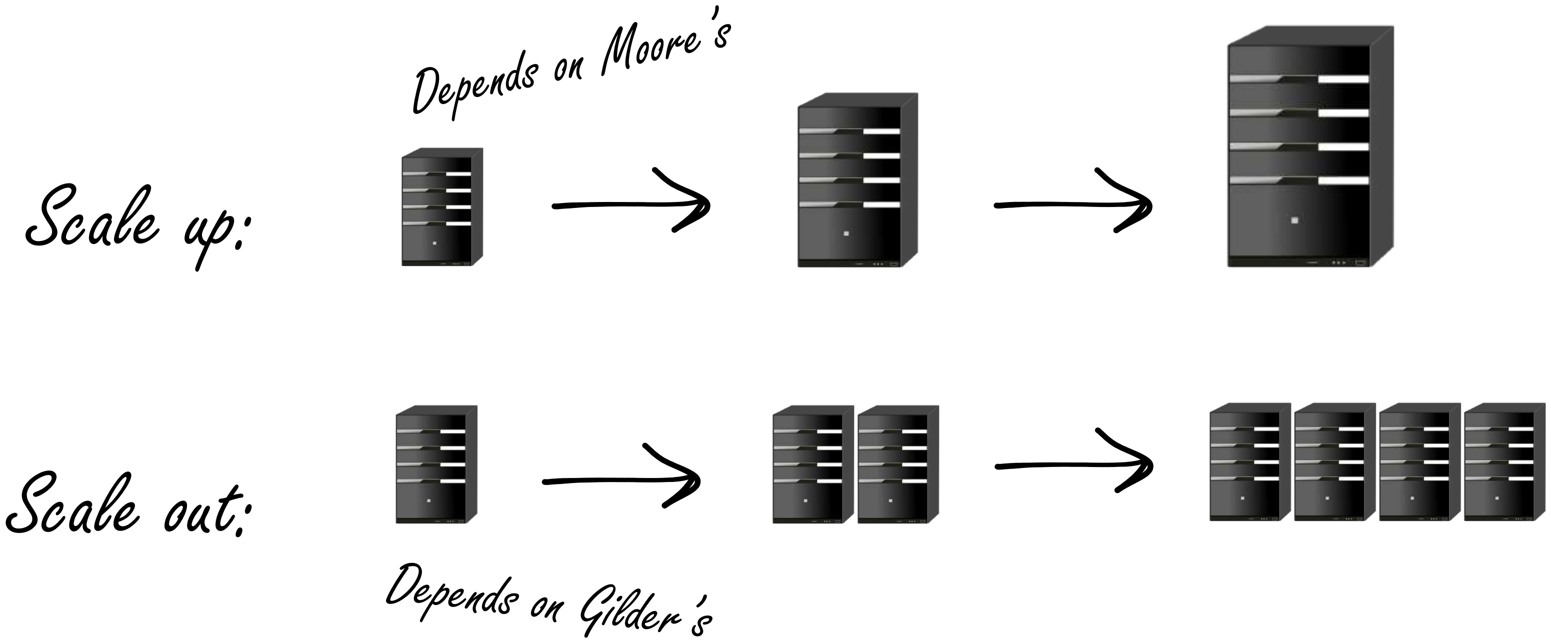
The Outer World

Scaling laws

What would you rely the most on?



Scaling laws



Scale up:

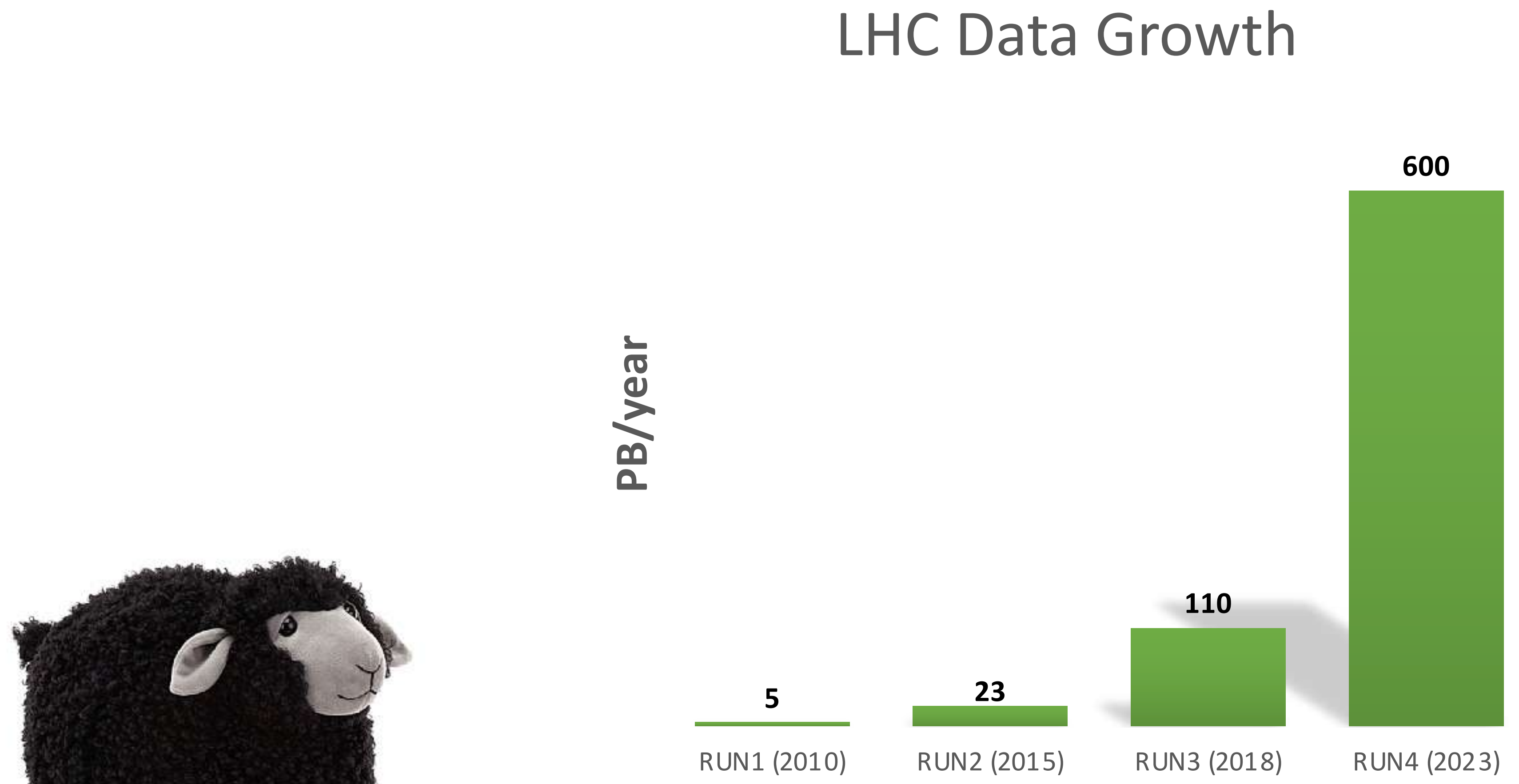
Depends on Moore's

Scale out:

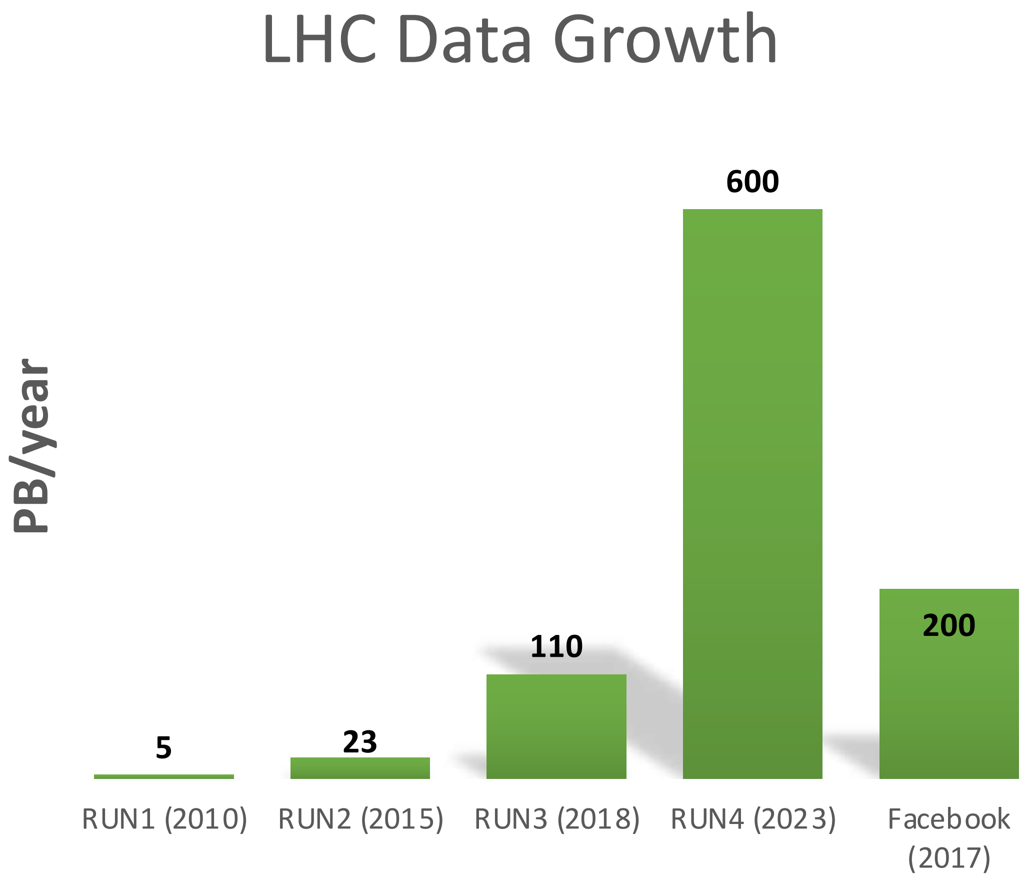
Depends on Gilder's

Nowadays, solutions which scale out are better than ones which scale up.
Things can change in the future, but we can even discover supersymmetry…

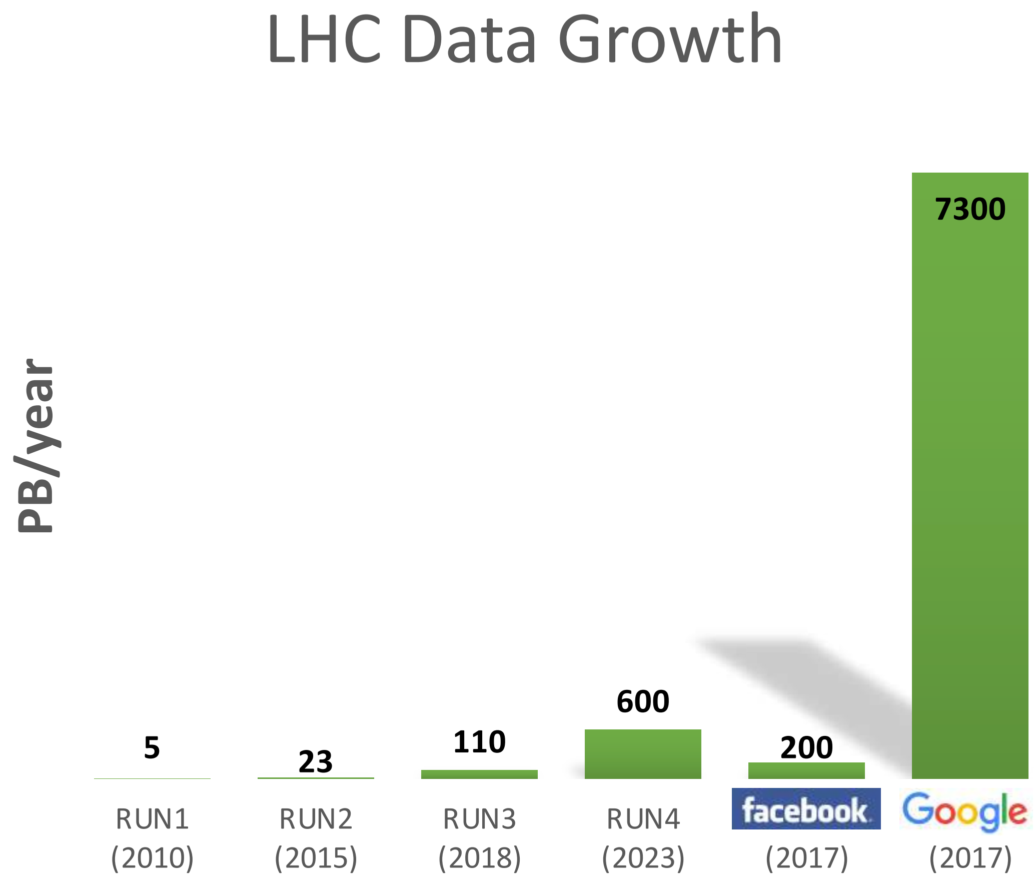
Are we a black sheep?



Are we a black sheep?



Are we a black sheep?



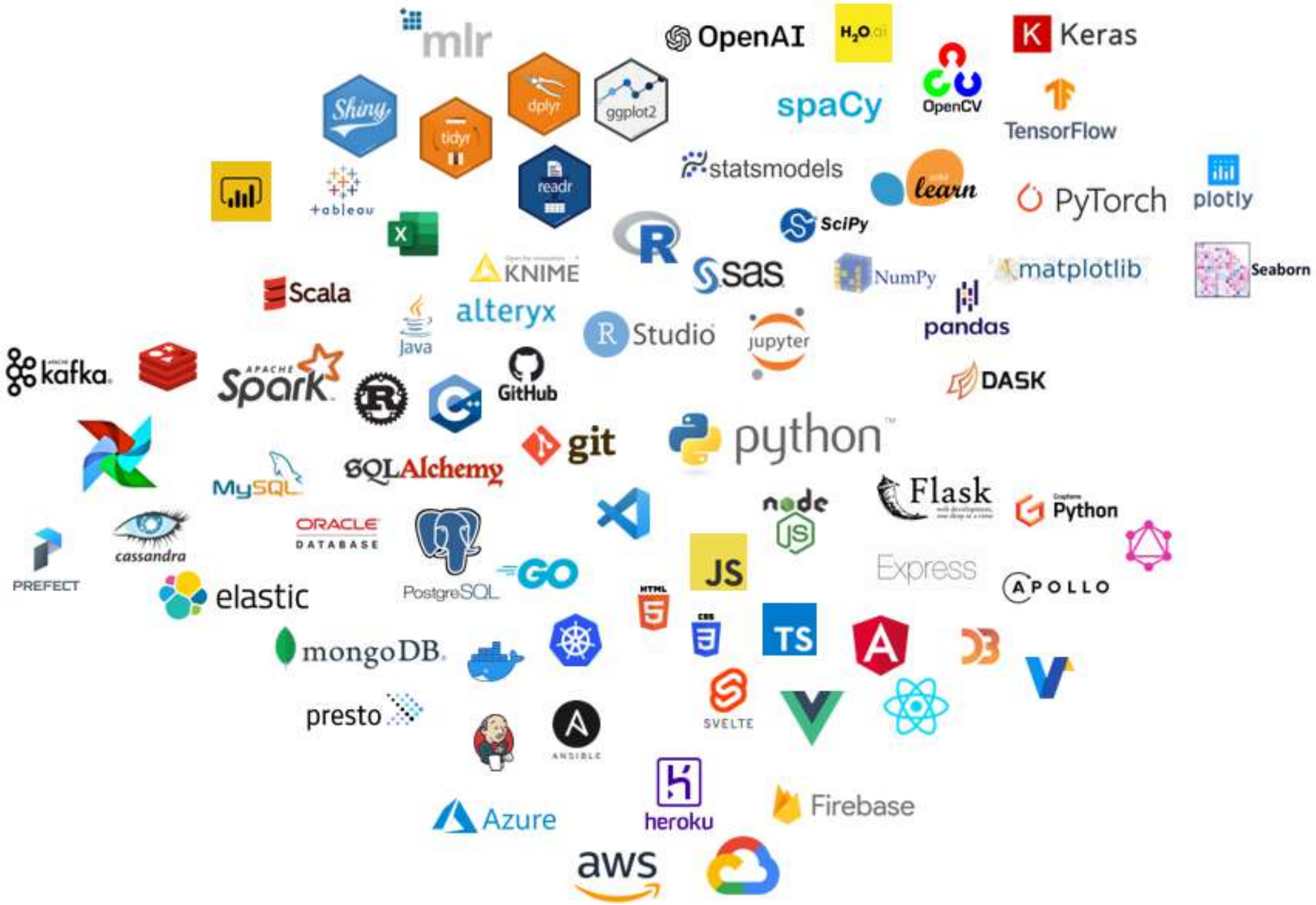
What can they offer?

Big tech companies are creating and maintaining open source projects.

These projects are extremely valuable and they're used by thousands of people actively developing (and fixing!) them.

Choose the right ones and use them as the backbone of your infrastructure.

Focus on the frontend and your users requirements and expectations!



One commandment…

DO NOT REINVENT THE WHEEL

The IT world has the right solutions for you.
Don’t be shy of relying on them, a solution with thousands of users is better than a self made one.

One commandment…

IMPROVE IT!

Either by participating to the development of existing solutions
or
introducing new ones only if “market” requires it.

Farewell checklist

Farewell checklist

- Gravitational waves are both a scientific and technical challenge
- A geographically worldwide collaboration requires an additional effort to create a coherent computing infrastructure
- There are three fundamental computing assets one must address: storage, software, computing
- Always try to decouple backend (aka what sysadmins prefer) from frontend (aka what users prefer)
- Try to adopt a reliable and affordable philosophy (no, not foolosophy despite Jamiroquai…)
- Stop hosting occasional services, focus on making them portable
- Scientists are not alone (anymore): don’t be shy of testing and improving somebody else’s wheel!

Thank you for your attention!